AUTOMATED VISUAL INSPECTION OF STEEL SURFACE, TEXTURE SEGMENTATION AND DEVELOPMENT OF A PERCEPTUAL SIMILARITY MEASURF

by

Prithwijit Guha

DEPARTMENT OF ELECTRICAL ENGINEERING

Indian Institute of Technology Kanpur

April, 2001

AUTOMATED VISUAL INSPECTION OF STEEL SURFACE, TEXTURE SEGMENTATION AND DEVELOPMENT OF A PERCEPTUAL SIMILARITY MEASURE

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Master of Technology

by

Prithwijit Guha

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
April, 2001

30-4-2001

1.

Certificate

It is certified that the work contained in the thesis entitled "AUTOMATED VISUAL INSPECTION OF STEEL SURFACE, TEXTURE SEGMENTATION AND DEVELOPMENT OF A PERCEPTUAL SIMILARITY MEASURE", by Prithwijit Guha, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Amitabha Mukerjee)
Associate Professor
Department of Computer Science & Engineering
Indian Institute of Technology, Kanpur

[illegible]

(Dr. Sumana Gupta)
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

April, 2001

To

My Parents

Acknowledgements

I would like to express my sincere gratitude to Dr. Sumana Gupta, Mr. Susmit Sen and Dr. Amitabha Mukerjee for their invaluable guidance and encouragement in completing my thesis work. I am specially grateful to Mr. Susmit Sen and Mr. Rajinder Rai for their help in designing and fabricating the necessary hardware, without which the project could not have reached its proper completion. I would also like to thank my friend Rajrup for being a great helping hand throughout the project. I am also very much thankful to all those friends of mine who have helped me in carrying out the psycho-visual tests without whose supports and advice, I couldn't have implemented a part of my work.

I would love to mention the names of my dear friends Kaushal, Prasad, Bajpai, Amitava, Prabhat, Aswini, Arghajit, Santanu, Arup, Sona, Rupen, Arindam, Rajarshi, Shamik, Tirtha, Himanshu and Abhishek, who have made my stay at IITK, an unforgettable experience.

Finally, I wish to acknowledge RDCIS for sponsoring our work and also the SAIL people for their assistance and support during my enjoyable stay for the project installation at Ranchi.

Abstract

In this thesis work we describe the prototype system developed for *Real Time Visual Inspection of Cold Rolled Steel Surface Defects*. The proposed system aims at detecting four classes of surface defects, viz. *Anneal Colour*, *Black Patch*, *Hole* and *Indentation Mark*. The prototype system hosts a simulated conveyor drive for keeping steel sheets in motion along with an imaging setup consisting of a well designed illumination system and an interlacing camera interfaced with Matrox Meteor II image acquisition card operated through a Pentium Class Processor. Content based image retrieval techniques have been used for detection and classification of surface defects. The image blocks are submitted as queries, which are processed for defect identification through Artificial Neural Network based Colour Histogram Feature Classification, Surface Modeling and Image Thresholding. The thesis also discusses algorithms for textural feature extraction and classification applied to supervised segmentation of multi-textured images, which could be extended to identify defects that are characterized by textures. More so, we propose a new measure of visual similarity developed on a perceptual framework. The new measure computes the indices of similarity (dissimilarity) between the query and the image database, thereby providing psycho-visually viable image retrieval results. We also discuss the possible application of the new measure to the problem of defect classification as an alternative procedure of content based image retrieval. Finally, the thesis discusses the possible future extensions to the prototype system along with suggestions to develop the high-speed industrial inspection system based on parallel imaging hardware.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The issue of Quality Control is an important aspect of today's highly competitive Industry One important way to improve the quality of the end product is to inspect the output of each manufacturing process However, Manual inspection of end products slows down the entire process as it becomes costly, time consuming and also may impact the effectiveness of human labor due to the hazardous atmosphere of industry

Therefore, the process of inspection is also to be automated ([1], [2]) and inspection results should be fed back to the upstream manufacturing process for improvement of product quality However, the Inspection system should be designed to be an efficient composition of human intelligence and experience along with the fastness of a machine This work deals with the approaches adopted to *Automated Visual Inspection of Cold Rolled Steel sheets for Detection and Classification of Surface Defects* and describes the Intelligent Prototype developed at the Robotics Lab, IIT Kanpur for the same

The content based imaging techniques have been applied for identification and localization of the surface defects The different defects are detected through Artificial Neural Network based colour classification, surface modeling and image thresholding The thesis also discusses the applications of texture segmentation and suggests the algorithms for classifying images rich in textural features More so, we develop an alternative measure of visual similarity based on a perceptual framework The results of implementation of the newly proposed measure have been discussed along with its possible extensions for applicability in steel surface defect classification

1.1 The Issues in Surface Defect Inspection

At present there are commercially available products which can detect the presence or absence of surface defects at reasonable costs However, this problem still remains an open research issue due to the difficulties faced during *Real Time Defect Detection*, *Identification* and *Localization* as well The major obstacles in this area are mainly due to the computational costs, lack of expert knowledge in defect feature selection or modeling, availability of proper defect samples etc The following discuss these issues in further details

A. **High Data Throughput:** A typical CRGO sheet is 1 to 3 m wide with a thickness of 1 to 5 mm These steel sheets of endless length move continuously on the conveyer belt at a very high speed of about 20 m per second Thus, it is a tough job for the inspection system to acquire and effectively process a high amount of data in a short amount of time

B. **Inter-class Similarity and Intra-class Diversity:** A single class of defect may vary widely in appearance and structure More so, the members of one class of defect may closely resemble to that of the other class So, the inspection system may easily get confused during the process of defect identification

C. **Large Number of Classes:** A typical defect identification scheme should deal with a large number of defect classes It is not unusual to deal with a few dozen of defects

D. **Non availability of adequate imperfection imagery:** Another very significant problem encountered during the design and development of the inspection system is the non-availability of adequate imperfect imagery for feature extraction and machine learning The hazardous environment of the steel industry hampers the collection of imagery data This problem is more acute due to the fact that the process of machine learning requires a very large amount of training data for proper identification of the high number of defect classes

E. **Dynamic Defect Populations:** Little changes and alterations in the manufacturing process may create an entirely new set of defect classes or may add up new types of defects to the existing ones The inspection system should be intelligent enough to adapt itself with the changing defect population Thus, the system should have the ability of on-line learning

The design of the inspection system [3] is determined by the first four factors, which defines the necessary hardware and the algorithms for proper data processing However, the fifth factor seems to be the most challenging one, which reflects the intelligence of the system in extending its capability with the changing environment

1.2 Existing Techniques of Defect Identification

The commercially available products approach the issue of surface inspection in a number of ways These methods are mainly based on the procedures of edge detection, profile analysis or multi-resolution image processing ([4], [5]) These procedures are briefly discussed in the following paragraphs

The process of **Edge Detection** ([4], [5]) assumes that the perfect steel sheet is smooth enough The steel surface image is filtered for removal of noises added up due to acquisition process The filtered image is then processed for multidirectional edge detection and proper thresholding for discarding noisy edges The edge map indicates the presence of defects, as a perfect steel sheet is enough smooth not to present edges The edge map or the segmented image provides the structural features of the defects, which are marked by matching them from a previously stored feature database The **main problem** of this process of defect identification is the fact that several defects may exhibit the same geometry, while the members of the same defect class can be structurally very different and thus they can't be distinguished by merely studying their structural appearance This is a case of **inter-class similarity and intra-class diversity** The best example is the case of Black Patch and Anneal Color, where both can occur with same geometrical appearance giving rise to structurally same edge maps, though the defects are of the very different classes

Another approach to defect identification deals with the **profile analysis** of steel surface images It is assumed that the perfect steel surface exhibits a smooth profile, whereas the same gets disturbed and a zigzag profile is presented during the presence of a defect The surface profile is estimated from the surface reflective power measured indirectly from the gray level [0 to 255 for 8-bit monochrome image] of the digitized image For the ease of calculation, it is assumed that the brightest point of the image has surface normal located on the axis of incident light This assumption provides a way for calculating the numerical relationship between digitized image and incident angle The incident angle gives an estimate of the surface profile [1] Though this procedure is efficient enough in detecting defects, it still exhibits poor performance in defect characterization

The third approach of defect analysis makes use of **smart sensors** and **cortical projection** In this method, the sensors function depending on the defect detection subsystem The analysis of the low-resolution image reflects the presence of any potential defect The presence of this potential defect alerts the detection subsystem through a feedback loop, which then makes the sensors to focus on the region of the potential defect The high-resolution image of the defective region is then analyzed The switching of sensing strategy is motivated by the desire to reduce data throughput This organization is particularly suitable for obtaining a cortical projection [3] of an image i e a geometrical transformation associated with human visual system Cortical projection simplifies the rotation and scaling effect of object in image plane

1.3 Content Based Imaging Techniques

The Content Based Imaging [6] is becoming increasingly popular in vision based applications due to its efficiency of reducing data dimensionality while retaining the distinct features of the data set In this method, a number of features are extracted from the data set by combining statistical characterization of the various linear or non-linear transforms of the data set The feature vectors so obtained are interpreted as points in the feature space where similar images produce feature vectors or feature points that cluster together in a certain region in the feature space Typically, the features of the query image are given to a *Content Based Image Retrieval* system, which hosts a pre-designed feature database, over which the system executes various classification algorithms to retrieve the most similar image(s)

Most of the classification algorithms assume the pre-condition for linearly separable clusters However, adding to the complexity of the classification process, there may be cases where clusters overlap sufficiently so as to confuse the classifier algorithm Thus, both the classification algorithm and its parameters are to be chosen properly so as to avoid the cluster overlap problems The feature extraction schemes are very often cascaded with Artificial Neural Networks [7], which classify the data sets in higher dimensions so as to get rid of the problem of cluster overlapping In the present work, we explain the performance of Artificial Neural Network based classification schemes applied to steel surface defects

The similarity of two images is often determined by some metric distance measure between the feature points – less the distance, more the similarity This scheme based on feature distance is called the *Minimum Distance Classification Algorithm* However, in most cases the validity of the distance measure method is questionable Reseaichers have refuted most of the metric axioms for measuring human perception of similarity or dissimilarity ([8], [9]) Tversky [8] and Santini [10] have proposed *Set Theoretic Similarity measures* to model human perception In this work, we also propose an index of similarity on the perceptual framework for the purpose of efficient image retrieval However, the newly proposed measure couldn't be applied for the surface inspection problem due to non-availability of proper defect images and its performance is validated on some simple Geometric figures

Texture is supposed to be one of the most important visual components and the Computer Vision community has performed several research works on analyzing the same Here we attempt the problem of texture classification and textured image segmentation using Gabor filters ([11], [12]) and propose a novel method of designing Gabor filters for *supervised texture segmentation* The performance of the algorithm is validated on multi-textured images formed from the Brodatz Texture database The problem of defect identification and localization is also taken care of from the viewpoint of texture classification Unfortunately, textural features do not characterize the defect classes attempted in this work and hence the approach is not implemented in the surface inspection problem

1.4 Overview of our Scheme

The present scheme focuses on the prototype developed at the Robotics Lab, IIT Kanpur for Detection and Classification of Steel Surface Defects Figure 1 1 shows a schematic block diagram of the prototype system

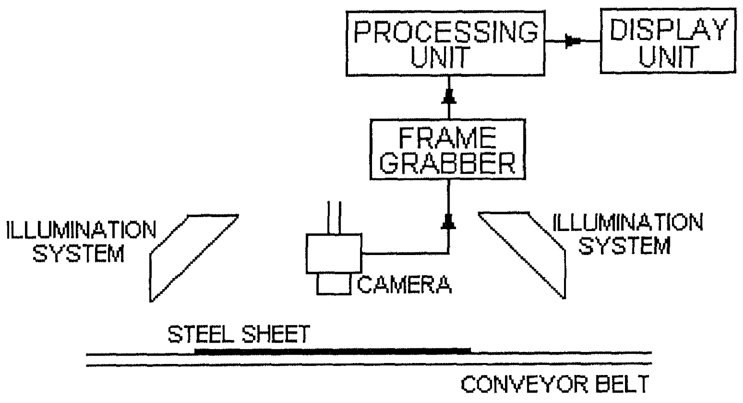


Fig. 1.1 : Schematic Block Diagram of the Prototype System

The prototype hosts an imaging system along with necessary hardware for simulating plant conditions The imaging setup consists of a camera unit, illumination system and computing hardware The plant condition is simulated by the conveyor belt unit, which sets the steel samples in motion The camera acquires the image of the steel sheet moving on the conveyor belt The illumination system helps in image acquisition by maintaining the proper level of brightness of the steel sheet The illumination sources are placed by trial and error so as to minimize the reflection from the glazy steel surface, which hampers the visual information of the digital image The acquired image is transferred to the processing unit through the Frame Grabber Card The processing unit executes the algorithms of defect detection, feature extraction and classification on the digital image thereby marking the defective areas of the steel surface which is shown to the user in the display unit In this work, we explore the operation of the prototype system in further detail and discuss the defect classification strategies involved

1.5 Thesis Organization

This thesis work is mainly focussed to Automated Visual Inspection of Steel Surface for Identification and Localization of Defects Besides, we discuss newly proposed methods for Supervised Texture Segmentation and computing Psycho-visually Viable Similarity Indices The thesis is organized in the following manner In chapter 2, we present the work on defect feature extraction and classification along with a overview and performance of the prototype system Chapter 3 focuses on designing Gabor filters for supervised texture segmentation and implementing the same for multi-textured images The newly proposed measure of psycho-visual similarity is dealt with in chapter 4 along with its application to simple geometric figures Finally, chapter 4 discusses the conclusions and possible future extension of the work

Chapter 2

Surface Defect Identification

In this chapter we present the imaging algorithms implemented for defect classification and discuss the performance of the prototype system developed This chapter is organized in the following manner Starting with a general overview of a typical classification system, we explain the procedures adopted for defect identification on that framework Finally, the descriptions of prototype hardware and system performance follow the algorithm discussion

2.1 Classification Systems

The identification of any object or a pattern involves the processing of its distinctive features The features play the role of abstract representation of any object As for example, we represent a flower by a number of features like its shape, odor and color These are the key features for classifying that flower and differentiating it from some other object Any object can have its physical, structural and mathematical features The sensory organs define the physical features of an object (eg touch, taste, color, odor etc) The structural features are defined by the geometry of the object (such as shape) However, the mathematical features are tough to compute (without some external computational aid) Obviously, for machine based classification applications, efficient recognition of physical features is tough to implement since the machines can't reach the higher complexity of human perception Thus, we go for extracting the mathematical features for representing the objects Examples of these types of features are their statistical characterization (may be in some transformed domain), Eigen values, Eigen vectors etc [13]

Figure 2 1 shows the block diagram of a typical pattern classification scheme In its first stage, the mathematical features of the input pattern are computed in terms of

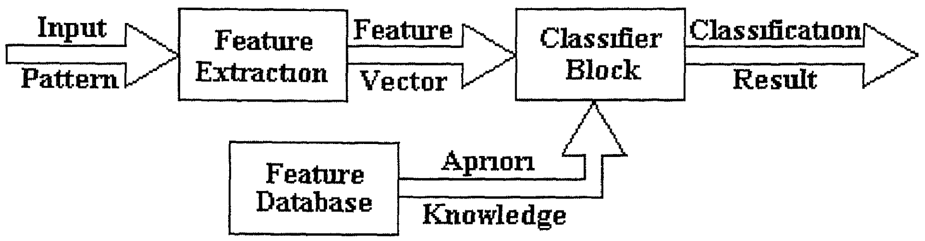


Fig. 2.1 : A Typical Classification Scheme

the statistical analysis of the pattern or its transform(s) These mathematical features form the feature vector, which represents the pattern as a point in the feature space of the set of all patterns (which have similar mathematical representations) The classifier stage identifies the pattern by using some apriori knowledge obtained from some previously stored feature database The feature identification can be performed by using minimum distance classifiers or through Artificial Neural Networks trained on a feature database

Our scheme deals with detection and classification of four classes of steel surface defects, viz Anneal Color, Black Patch, Hole and Indentation Mark Anneal Colour and Black Patches are characterized by their distinctive colours from the non-defective steel parts Thus, the colour features of these types of defects are essential for identifying them On the other hand, Holes and Indentations Marks are localized discontinuities or depressions on the steel surface and their presence can be detected by the characteristics of the surface illumination at the defective parts The following sections describe the different approaches adopted for selecting features, designing feature extractor blocks and classifier stages

2.2 Color Histogram Feature Extraction

The colour is the key feature in case of identifying Black Patch and Anneal Colour, as they are rich in colour information Anneal Colours are markings usually found along the coil edges in hood annealed material whose colour can vary from light brown to deep blue On the other hand, Black Patches are dark black stains found on the steel surface as bands or clusters Apart from these two, the perfect steel sheets are characterized by their glaze, which appears almost white in the digitized images Thus, this wide variability of colour among the different defect classes and the perfect steel helps in the proper differentiation and identification of these classes

The pixels of coloured images are typically characterized by the three colour components viz the Red, Green and Blue (RGB Colour Space convention), which together produces the original colour Thus, to extract colour features, we need to process the features of the individual colour components In addition to the RGB colour components, we also consider the monochrome value (between 0 to 255) of the pixel as a feature The gray scale value (g) is obtained from the colour components [Red (R), Green (G) and Blue (B)] according to the following equation -

$$g=\sqrt{\frac{R^2+G^2+B^2}{3}} \qquad (2\ 1)$$

However, pixel-wise colour classification will be a slow process and is particularly unsuitable for real-time image processing Thus, we go for block feature extraction, where the image is subdivided into a number of non-overlapping blocks The colour features of each block are computed to mark the block to be belonging to a particular defect region or non-defective otherwise We typically choose a block size of 50 x 50 sq pixels and its features are extracted from the colour histograms of the block Figure 2 2 shows the histograms of the red, green, blue and the gray level components of the 50 x 50 blocks of Anneal colour (A), Black Patch (B) and Perfect Steel (C) It is evident from figure 2 2, that the histograms of the three different classes have wide variability in terms of their shape and location

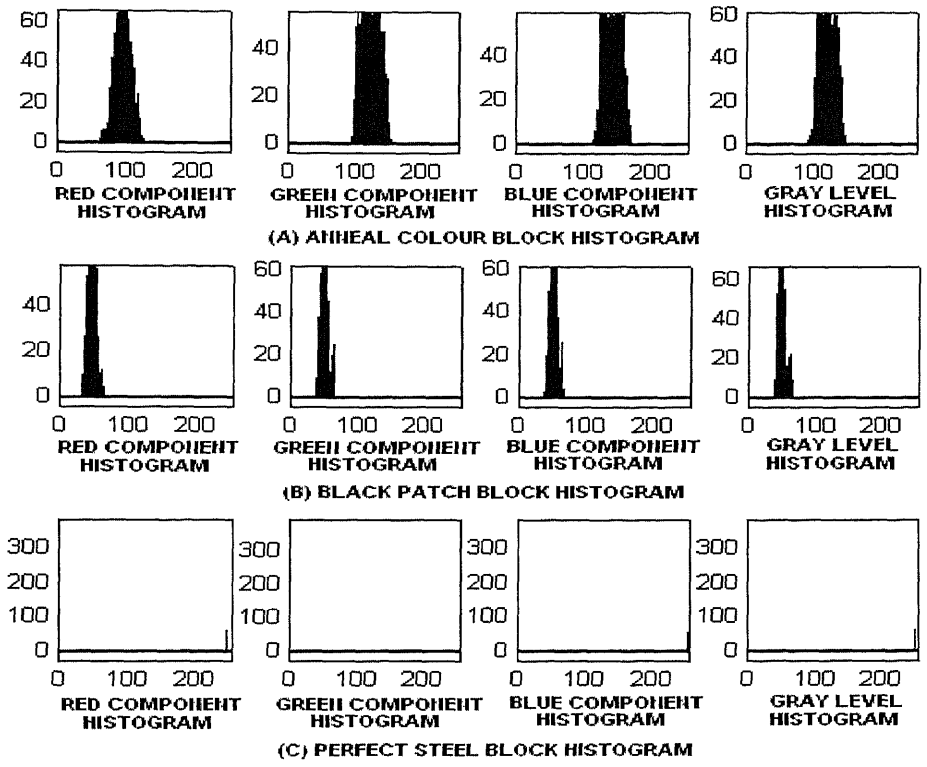


Fig. 2.2 : Colour Histograms of (a) Anneal Colour, (b) Black Patch and (c) Perfect Steel

Hence, the feature vector of the block is constructed from its colour histogram statistics The weighted mean, variance and third moment are computed as histogram features of each colour component Thus, the feature vector (F) of any block is formed as -

$$F = [M_{red}, V_{red}, S_{red}, M_{green}, V_{green}, S_{green}, M_{blue}, V_{blue}, S_{blue}, M_{gray}, V_{gray}, S_{gray}] \qquad (2\ 2)$$

Where, M, V and S denote the weighted mean, variance and third moment respectively, suffixed with the respective colour components

Thus, each block is represented by a point in the 12 dimensional block feature space It is expected that the blocks form three distinct clusters in the feature space In that case, we could classify any feature point by merely computing the minimum distance of the point from the cluster centres Unfortunately, this clustering is not simple enough and it gives rise to highly overlapping clusters (not linearly separable) in the feature space Thus, we go for Artificial Neural Network (henceforth denoted as ANN) based classification strategies, which classifies feature points in higher dimensional spaces by forming convex hulls, thereby separating the overlapping clusters The following section briefs the concepts of ANN based classifiers

2.3 Artificial Neural Network based Classifiers

The applications of Artificial Neural Networks [7] have become increasingly popular in the fields of pattern recognition ([6], [14]) and system approximation for their successful performance within allowable error levels The Neural Networks approximate the input-output relationship of any system by a set of network parameters and perform within some tolerance when trained to desired error level The most popular neural network in use is the Multi-layer Perceptron (MLP), which learn the training patterns under a supervised framework using the error backpropagation algorithm [7]

Though, the operation of a MLP is quite simple, Radial Basis function Neural Networks (henceforth denoted as RBFNN) have been applied more successfully in problems of pattern classification [7] The RBF takes a different approach to learning [15] by approximating the surface in the multi-dimensional space that fits the best to the training data The following sub-sections discuss the RBF Networks and its application to pattern classification

2.3.1 The RBFNN Classifier Architecture

The Pattern Recognition community has frequently applied the classifiers based on RBF Neural Networks as it efficiently estimates the surface separating the pattern classes thereby leading to successful pattern classification [7] A simple RBFNN has a typical three-layered architecture consisting of the input layer, the hidden layer hosting the Radial Basis Functions and the output layer Figure 2 3 shows a schematic sketch of a typical RBFNN having N number of inputs, P Radial Basis Function

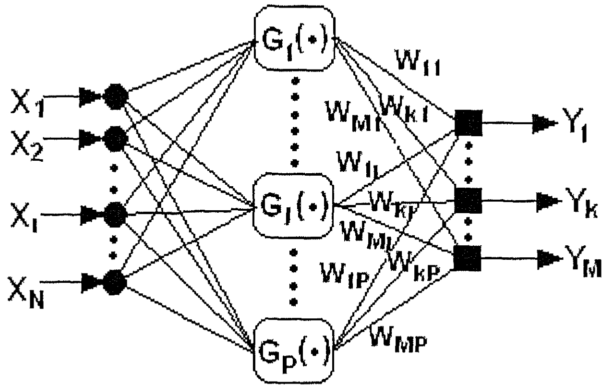


Figure 2.3: Radial Basis Function Neural Network Architecture

nodes in the hidden layer and M outputs The network computes the Euclidean distance of the input vector from each of the Radial Basis centers, which then undergoes a nonlinear transformation by the Radial Basis Function to generate the hidden layer output The network output is simply a weighted sum of the hidden layer outputs The RBFNN implements a mapping of input vector X to output vector Y as $f : \mathbf{R}^N \rightarrow \mathbf{R}^M$ according to,

$$y_k = \sum_{j=1}^{P} W_{kj} G_j\left(\left\| X - C_j \right\|\right) \qquad (2\,3)$$

Where, $X \in R^N$ is the input vector, $G_j(\,)$ is the Radial Basis Function from $\mathbf{R}^N$ to $\mathbf{R}$, $\|\ \ \|$ denotes the Euclidean distance, W_{kj} ($1 \le j \le P$ and $1 \le k \le M$) are the weights connecting the k'th output node to the j'th Radial Basis Function node, $C_j \in R^N$ is known as the j'th RBF centers, and P is the number of centers [7]

Haykins [7] describes a number of choices for the Radial Basis Function G, the most popular being the multi-dimensional Gaussian We use a further simplified form of the multi-dimensional Gaussian by neglecting the off-diagonal terms of the covariance matrix We take the Radial Basis Function as -

$$G(X) = \exp\left(-\frac{1}{2} \sum_{i=1}^{N} s_i^2 (x_i - c_i)^2 \right) \tag{2 4}$$

Where, s_i^2 are the inverses of the variances of the Gaussian s_i determine the spreads of the RBF along the different axes Higher value of s_i indicates a peaky Gaussian around c_i and less the value of s_i, flatter is the shape of the RBF

The RBFNN classifier [14] is a further extension to the RBFNN, where we look for the output node having the maximum activation and label the input to be of the class as indexed by the output node number As for example, consider the case of classifying M patterns from their N dimensional feature vectors The RBFNN outputs are computed according to equation (2 3) The input vector is labeled to be of the p'th pattern class if $Y_p > Y_k$, $\forall k = 1,2, \quad ,M$ and $k \neq p$

2.3.2 Network Training

The functioning of the RBFNN classifier depends solely on the network free parameters The network parameters, that is, the number of hidden layer nodes, RBF centers and spreads and the weights are to be chosen properly for optimized performance All these parameters are highly dependent on the training data The training procedure tunes these parameters by minimizing some error goal For designing the RBFNN classifier, we go for discriminative training, where the parameters are tuned for maximum discrimination of the training patterns of different classes

Consider the case of RBFNN classifier design for M patterns from their N dimensional feature vector Let, there be T number of feature vectors or training patterns available for network training The k-means-clustering algorithm [13] (described in Appendix A in details) is executed iteratively on the training data set to find out the maximum number of clouds where the data are clustered These centroids of these are used as the initial values for the RBF centers and the number of Radial Basis Functions (P) is the number of such clouds The variances of the vectors of each cluster are computed and their inverses serve as initial estimates of the RBF spreads For the feature vector of the p'th class we set the desired output vector D as,

$$\begin{aligned} d_k &= 1 , k = p \\ &= 0, 1 \le k \le M \text{ and } k \ne p \end{aligned} \tag{2 5}$$

Now, for initializing the weight matrix W, we compute the hidden layer outputs for all the T training patterns and form the matrix H of size P x T Similarly we form the matrix Q of size M x T consisting of the desired output vectors arranged in columns The initial weight matrix W (M x P) is computed from the relation,

$$W = QH^T(HH^T)^{-1} \tag{2 6}$$

For minimizing the training error, we compute the pattern error $\zeta(t)$ for the t'th training pattern as,

$$e_k = d_k - y_k \tag{2 7}$$

$$\zeta(t) = \frac{1}{2}\sum_{k=1}^{M} e_k^2 \tag{2 8}$$

Where, y_k is the network output computed by equation (2 3)

If the network parameters are optimized with respect to each pattern error, the procedure is called pattern mode training However, the parameters can also be adjusted by optimizing the global error over the training data set, which is the batch mode training The batch mode training allows a better estimate of the training error and is also faster than the pattern mode For training in batch mode, the pattern error is computed for each training data and the global error E is simply the average of these pattern errors computed as -

$$E=\frac{1}{T}\sum_{t=1}^{T}\zeta(t) \qquad (2.9)$$

The network-training objective is to minimize E [15] The parameters are updated till the required error goal is reached The parameters of the n'th iteration are updated for the (n+1)'th iteration according to the following equations (Appendix C)

$$c_{ij}(n+1)=c_{ij}(n)+\eta_c\left\{\sum_{k=1}^{M} e_k w_{kj}(n)\right\}\left[h_j\{x_i-c_{ij}(n)\}s_{ij}^2(n)\right] \qquad (2\ 10)$$

$$s_{ij}(n+1)=s_{ij}(n)-\eta_s\left\{\sum_{k=1}^{M} e_k w_{kj}(n)\right\}\left[h_j\{x_i-c_{ij}(n)\}^2 s_{ij}(n)\right] \qquad (2\ 11)$$

$$w_{kj}(n+1)=w_{kj}(n)+\eta_w e_k h_j \qquad (2\ 12)$$

Where, η_c, η_s and η_w are the respective learning rates for RBF center, RBF spread and connection weights and h_j is the output of the j'th Radial Basis Function The learning rates for the RBF centers, spreads and connection weights should be different The RBF parameters (centres and spreads) should be learned at a slower rate as compared to that of the weights The learning rates should be further reduced after a certain number of iterations where the rate of change of error slows down or undergoes a change of sign [7]

The RBFNN steel surface image block classifiers are designed by the same procedure discussed above The block histogram features are extracted from the sample images by the procedure as mentioned in section 2 2 Along with the training procedure, the classification performance was also evaluated with the current set of

RBFNN parameters The learning rates for the RBF parameters are varied manually between 0 001 and 0 1 and that for the weights are varied between 0 1 and 0 9 according as the rate of change of training error The training is stopped if either the error goal reaches 0 001 or if the maximum feature misclassification (on the training set) becomes lower than five percent

2.4 Results of Coloured Defect Classification

The previous sections have discussed the feature vector selection and RBFNN based classification of defective regions The procedure is tested on a few steel surface images with Black Patches and Anneal Colours Figures 2 4 (a) and (b) show the results of defect region detection through colour histogram feature extraction and classification The results shown here are trained and tested on static image segments

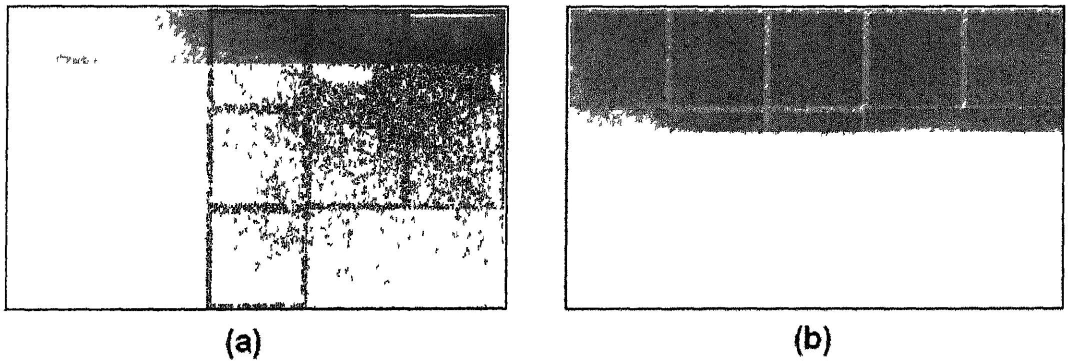


Fig. 2.4 : Block Labeled Images of (a) Anneal Colour and (b) Black Patch

In the practical scenario, the images get distorted due to motion blur However, the patch type defects like Black Patch or Anneal Colour doesn't get much affected due to blurring as they are spread over regions There we train the RBFNN classifier on the training set formed of image segments grabbed in both static and moving conditions Section 2 8 shows the results of motion blurred defect image classification

2.5 Detection of Indentation Mark

The ındentatıon marks are localızed depressıons (swellıngs) on the plane steel surface caused by protrusıons on defectıve rollers Thus, these types of defects ıntroduce some form of curlıness on the metal surface, whıch serve as a dıstınctıve feature of the defect Hence, for characterızıng ındents we go for modelıng the surface of the defectıve regıons Typıcally, any type of surface (curved or plane) could be modeled by usıng hıgher order polynomıals However, modelıng hıgher order polynomıals ıs a computatıon ıntensıve job and ıs not possıble for real tıme ımplementatıons More so, swellıngs or depressıons beıng nearly quadratıc surfaces, we go for two-dımensıonal second order polynomıal modelıng, whose parameters serve as the surface features The followıng sub-sectıons brıef the mathematıcal detaıls of surface modelıng, algorıthm ımplementatıon and the results of detectıng ındents

2.5.1 Quadratic Surface Modeling

A two dımensıonal second order polynomıal $S_q(X,Y,P)$ [$(X,Y)\in R^2$ and $P=\{A, B, C, D, E, F\}$, the set of polynomıal parameters] ıs expressed as -

$$S_q(X,Y,P) = AX^2 + BY^2 + CXY + DX + EY + F \qquad (2\ 13)$$

The task of modelıng a two dımensıonal surface S (defıned over [a,b]x[c,d], $a,b,c,d\in R$) by the quadratıc surface $S_q(X,Y,P)$ ınvolves fındıng the optımal parameter set P*, such that the average modelıng error E ıs mınımızed, where E ıs gıven by,

$$E = \frac{\iint\limits_{[a,b]x[c,d]} \left\| S - S_q(X,Y,P^*) \right\| dXdY}{\iint\limits_{[a,b]x[c,d]} dXdY} \qquad (2\ 14)$$

For the purpose of detectıon of ındents, we consıder surface modelıng of blocks of dımensıons 50 x 50 sq pıxels However, computatıon for so many pıxel positions is also time consuming and thus we compress the block to the size of 25 x 25 sq pixels Each pixel value of the compressed block is computed from the average of four neighboring pixels in the original block The process of averaging performs a smoothing operation and benefits the process by reducing the level of noise

A rectangular co-ordinate system is considered for the surface-modeling problem, which have its origin at the center of the 25 x 25 block Thus, the surface to be modeled is given by the pixel values of the block defined over the pixel position domain of [-12,12] x [-12,12] However, this co-ordinate system is different from the block co-ordinate system, which has its origin at the top-left corner of the block and points to block elements in terms of block-matrix rows and columns The (p,q)'th pixel value of the block co-ordinate system is related to the (X,Y)'th surface value as,

$$p = Y - 12 \tag{2 15}$$

$$q = X + 12 \tag{2 16}$$

To solve for the optimal parameter set P*, we equate the quadratic expression S_q given by equation (2 13) to the block pixel values Thus, the equation for the (X,Y)'th value of the block matrix M is formed as -

$$AX^2 + BY^2 + CXY + DX + EY + F = M(p+12, q-12) \tag{2 17}$$

However, for the 25 x 25 block, we can form 625 such equations from which we solve for the six unknowns A, B, C, D, E and F To express in matrix notations, we have to solve for the equation,

$$VP = T \tag{2 18}$$

Where, the dimensions of the matrices V, P and T are 625 x 6, 6 x 1 and 625 x 1 respectively Each row of matrix V is formed as $[X^2\ Y^2\ XY\ X\ Y\ 1]$ and that of T as [M(p+12,q-12)] and P is the column vector $[A\ B\ C\ D\ E\ F]^T$as shown in equation (2 17) Since, the matrix V is not a square matrix, taking the inverse is not possible for solution of P Thus, we go for the operation of pseudo-inverse, by which we solve for P from equation (2 18) as -

$$P = (V^T V)^{-1}(V^T T) \qquad (2\ 19)$$

Once the polynomial parameters are determined, we get a fairly well idea of the surface profile from them In the quadratic expression of equation (2 13), the coefficients of the second order terms A and B (X^2 and Y^2) determine the surface curliness The coefficient C (XY) determines the orientation of the curly surface and

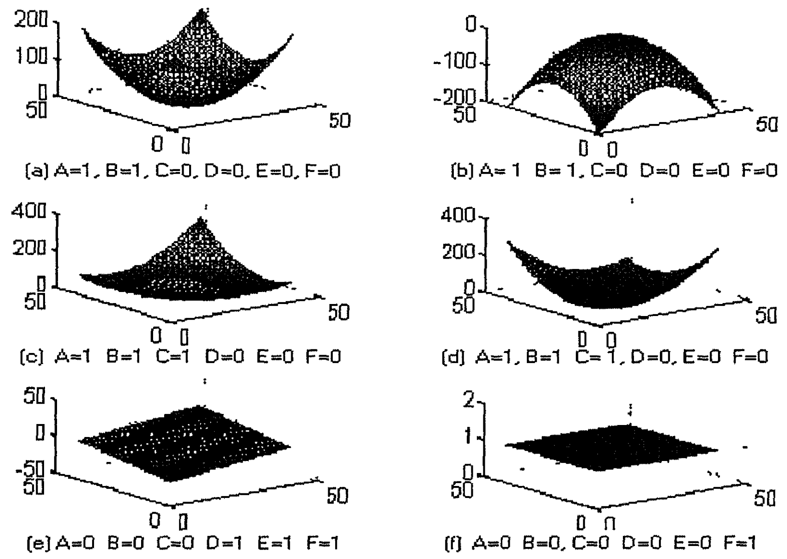


Fig. 2.5 : Effect of Polynomial Coefficients on Surface Profile

D, E and F are only the linear components, F being the surface datum Figure 2 5 shows the effects of the polynomial coefficients on the surface profile It is seen that negative or positive values of (A, B) determine the swelling or depression More so, C determines the orientation of the quadratic surface Thus, whatever be the type (swelling or depression), position or orientation of the indent, it will be well captured by the polynomial parameters The surface curliness being best estimated by the values of A and B, we select their absolute values (to cover both Swelling and Depression) as the features of presence of indent

2.5.2 Results of Indent Detection

The indent detection algorithm, as described above has been run on a number of test blocks having different surface profiles (swellings, depressions and even planes) Since, the magnitudes of A or B reflect the surface curliness, we choose a suitable threshold for the polynomial parameters by trial and error method It is found that the

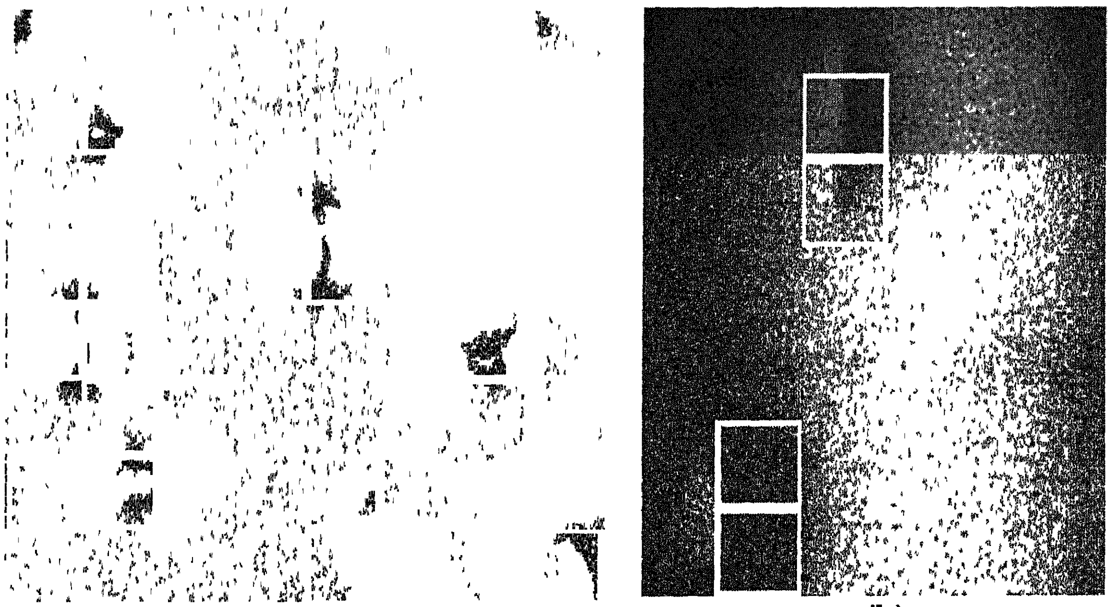
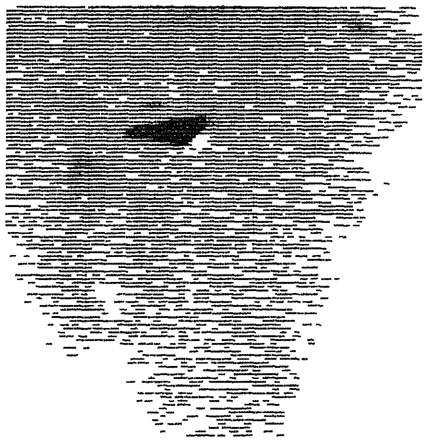

(a) (b)

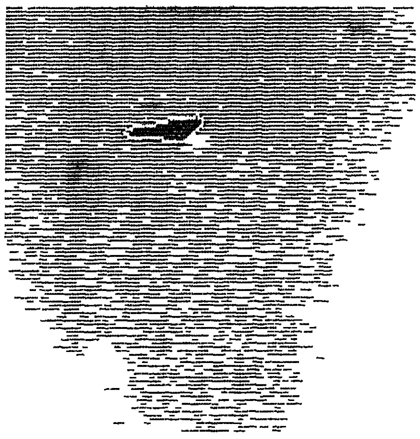
Fig. 2.6 : Detection of Indents (White margins enclose the defective region)

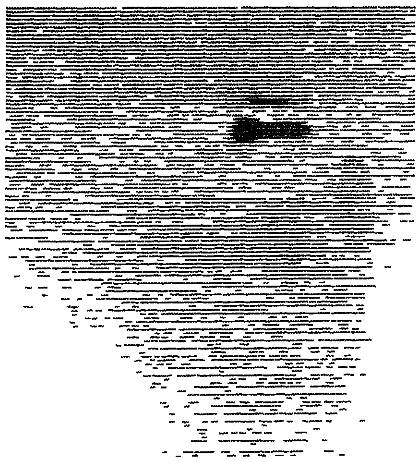
presence of an Indent is very well determined if we use a threshold of 0 05 If the magnitude of either B or A exceed this threshold, the block is marked to have an Indent Figure 2 6 (a) and (b) show the results of indent detection, where the defective blocks are marked with white margins Though the images [2 6 (a) and (b)] were taken under different conditions of illumination, the same threshold is found to work well for both Since, this algorithm works on surface modeling, it is only dependent on the surface profile and is quite independent of illumination

2.6 Detection of Hole

Detection of through holes on the steel surface requires the knowledge of the background, as it becomes visible through the material Generally, the background is black due to the colour of the conveyor belt surface Thus, the visibility of a dark background through the material characterizes the presence of a hole Hence, the process of hole detection involves the thresholding of the image (in gray scale) by a suitable value below which the pixels represent the dark background region This threshold is dependent on the imaging system and is to be tuned manually with respect to the camera position and illumination


(A) Original Image


(B) Hole Detected Image


(C) Original Image

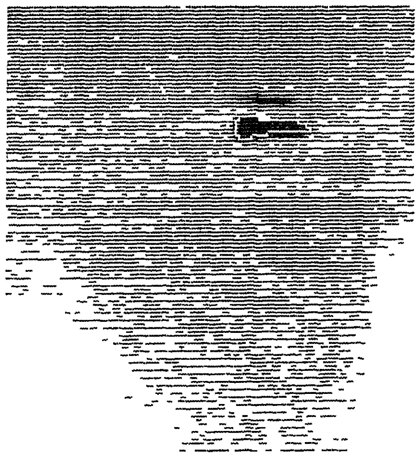

(D) Hole Detected Image

Fig. 2.7 : Results of Hole Detection

The image is subdivided into a number of 2 x 2 blocks The average pixel value of each block is computed and is compared with the set threshold Blocks having average intensity less than the threshold are marked to be belonging to the dark background region Figure 2 7 shows the results of implementation of the hole detection algorithm The algorithm is executed on two different images acquired under similar imaging conditions and the threshold is set to 90 (Where, pixel intensity value may vary between 0 to 255)

So far we have discussed the feature extraction strategies involved in surface defect detection and classification The results of algorithm implementation are also shown in the relevant sub-sections These results are computed from MATLAB simulations [16] performed on static steel surface images However, in the practical case, we have to process the images of the steel sheets grabbed in motion This result to motion blurs leading to loss of visual information, which may be due to the slower camera response compared to speed of moving steel sheets There, the algorithm parameters need to be tuned for optimal performance along with sufficient image preprocessing The next section explores the operation and performance of the prototype system in details along with practical implementation of the algorithms

2.7 Inspection System Overview

The Online Surface Inspection System aims at detection and classification of visual surface defects of CRGO (Cold Rolled Grain Oriented) steel sheets at very high speeds Typically, this speed varies from 20m/sec to 30m/sec The inspection system at such high speeds need to be operated with a parallel processing architecture (for reducing computational overheads), typically accompanied with a number of high speed cameras (to increase accuracy) The inspection system prototype developed at the Robotics Lab, IIT Kanpur is however a single processor system, operating with a much slower camera, performing on steel sheets moving at the rate of 1m/sec The plant level implementation of the high-end inspection system is only a scaled up version of the developed prototype

The prototype system mainly constitutes of two distinct parts, viz The Hardware setup and the Software Modules The following sub-sections discuss the operation of the hardware setup and performance of the software modules of the prototype system in further details

2.7.1 The Hardware Setup

The basic requirements of any real-time imaging system essentially consist of a well-designed illumination system along with cameras and data acquisition cards interfaced with an efficient computing hardware The imaging system of the prototype deals with real-time image acquisition and processing The steel sheets are kept on a moving trolley powered by a constant speed motor drive The imaging setup is constructed using a Watec 202 camera interfaced to a Pentium PC through the Matrox Meteor II image acquisition card An optimal level of visual information is achieved by maintaining a more or less constant level of illumination The PC functions in triggering image acquisition along with the processing of digital images The following subsections provide short descriptions of the different parts of the prototype system hardware

2.7.1.1 The Conveyor Belt Drive

In the practical plant situation, continuous steel sheets are carried away by conveyor belt drives and the inspection system operates on the steel surface images grabbed in motion For the prototype demonstration system developed at IIT Kanpur, a simple conveyor system, consisting of a 1ft x 1 5-ft trolley on aluminum tracks has been used The trolley is pulled back and forth by a cable attached to pulleys The pulleys are in turn activated by a gear and motor arrangement The main drive gear have teeth on half the circumference, while the remaining part of the circumference is plain and does not engage while in motion The main gear wheel thus engages on toothed gear wheels on either side of it as it rotates continuously In the process it energizes one set of pulleys in one direction and another set of pulleys in the opposite direction, thus resulting in to and fro movement of the trolley The system is energized by a 1 H P motor, rotating at 1450 rpm This rpm is reduced by a worm gear of ratio 1 40 to 36 25 rpm This is further reduced through a set of spur gears of ratio 1 3 to 12 083 rpm Since the main drive gear is larger than the driven gears by a ratio of 4 1 this results in a final rpm of 48 33 for the pulleys i e 0 8055 rotations per second Thus to achieve a trolley speed of 1 m /sec , pulleys of diameter 0 256 meters have to be used In practice, pulleys of inside diameter of 20 cm have been used so that the trolley starts at a slower speed and as the cable gets wrapped inside the pulley, the effective diameter increases and the trolley accelerates to the desired speed of 1 m /sec Figure 2 8 shows a schematic diagram of the system and figure 2 9 is the photographic view of the conveyor

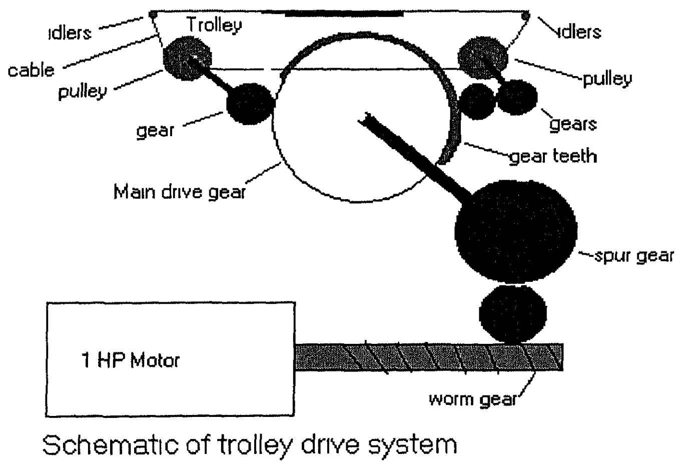


Figure 2.8 : Schematic Diagram of the Trolley Drive System

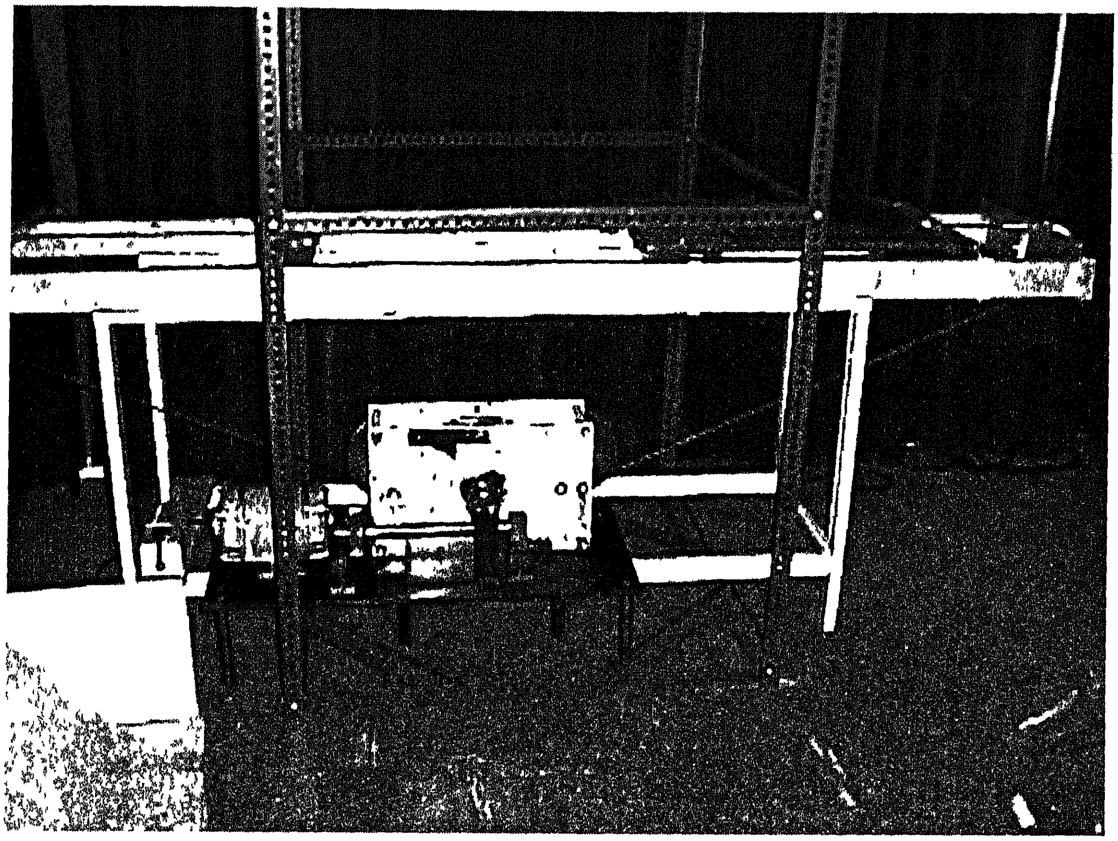


Figure 2.9: Photographic view of the simulated conveyor.

2.7.1.2 The Illumination System

The basic philosophy of image processing relies on one's visualization of the same Visual Information gathered from an image solely depends on its illumination conditions Both the extremes of illumination, brighter or darker, tend to reduce the visual information More so, each image should have its own illumination conditions to have maximum information content Thus, the level of brightness needs to be adjusted optimally [17]

The illumination system used for the prototype demonstration system consists of two banks of 3 metal halide lamps each of 250 watts The lamps are powered by 3-phase AC voltage at 230 Volts with respect to neutral Each phase lights diagonally opposite lamps to ensure uniform brightness at all instants of time Each bank of lamps is covered with a layer of frosted glass to ensure uniform glare-free illumination The lamps are angled upwards to ensure that there are no reflections from the steel surface Steel, especially perfect steel surface is extremely reflective and results in saturation of the CCD device in camera, reducing the information content in the image To ensure uniform non-reflective illumination, an enclosure of louvered steel sheets with black jute backing and black paint has been used Figure 2 10 shows a photographic view of the illumination system

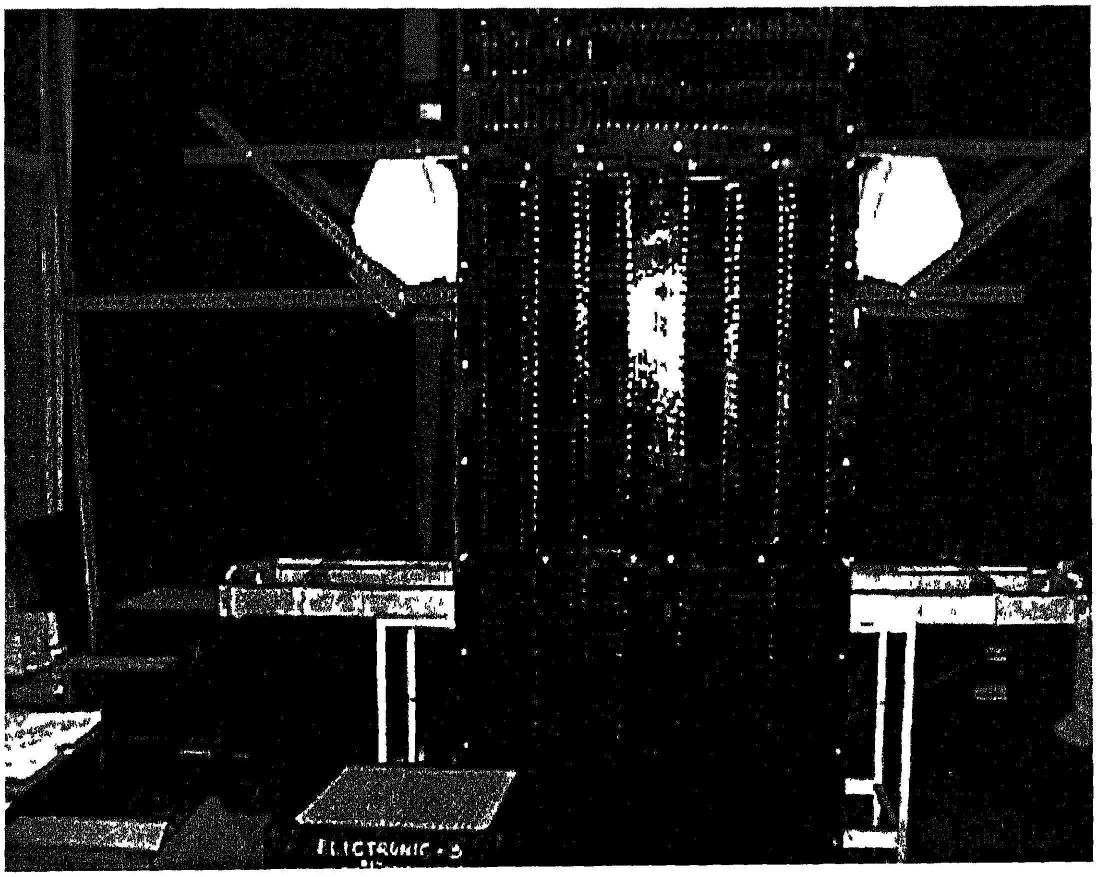


Figure 2.10 : Photographic view of illumination system. The two white metal enclosures house the lamps, angled upwards for diffused illumination.

2.7.1.3 The Data Acquisition Setup

The fundamental task done by the Data Acquisition Setup is grabbing frames via the CCD (Charged Coupled Device) camera interfaced to the host PC through the Matrox Meteor II card [15] A detailed diagram and explanation of the Matrox Meteor II card is given in figure 2 11 As shown in the figure, the frame grabber card mainly consists of the following components -

- Analog Input Selector
- Filter
- Video Decoder
- VIA or Video Interface Asic

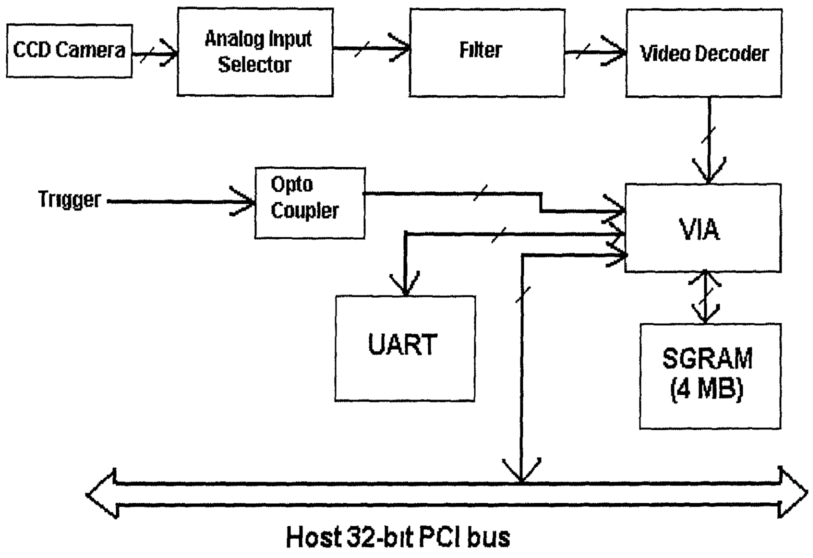


Fig. 2.11 : Block Diagram of Matrox Meteor II Image Acquisition Card

Video signal of camera is routed via Analog Input Selector to the filter, which is used to limit high frequency noise and aliasing effects at the input of the Video Decoder The Video Decoder transforms analog video signals to digitized component video The image is grabbed by the VIA either in single frame or in continuous acquisition mode according to the trigger input The VIA can also remotely control a camera, or a motion control unit or remotely communicate with a Program Logic Controller (PLC) via the UART (Universal Asynchronous Receiver Transmitter) The grabbed image is stored in the SGRAM and also placed in the PCI bus The red, green and blue components are separately extracted in three different parts of the host memory from the buffer in the SGRAM on which the processing is done

2.7.2 The Software Modules

The Software Modules are actually the algorithms coded into computing instructions that run on the Inspection System Hardware Setup for efficient detection and classification of steel surface defects The software section constitutes of modules for acquisition, processing and display of image data The software functions in a loop - acquiring frame data, processing for defect detection, displaying the results and waiting for the next frame to process and so on Figure 2 12 shows a block diagram representation of the whole software section The following subsections explore the functioning of the different blocks in further details

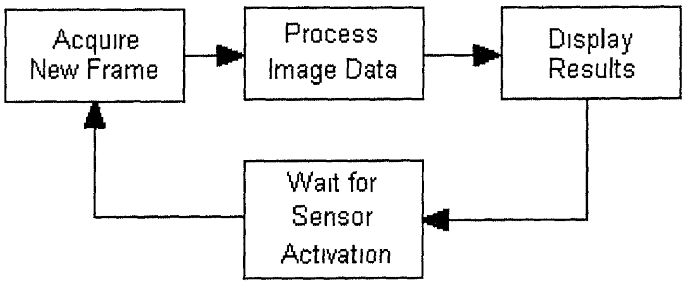


Fig. 2.12 : Schematic Representation of Software Functioning

2.7.2.1 Image Acquisition

The prototype system does not host an actual conveyor belt drive and thus we don't get continuous steel sheets An infrared, non-contact optical sensor is used to detect the correct position of the trolley, which gets triggered whenever the trolley comes just below the camera This sensor activation is tracked for image acquisition at the moment when the steel sheet on the trolley is just below the camera

The sensor output ıs connected to the parallel port of the computer (connector pın 10, sıgnal ACK, bıt 6 of prınter status port X79, where X ıs 3 for LPT1 or 2 for LPT2) Thus the status of the sıgnal ıs read by BIOS ınterrupt, INT 17H, functıon 2 Bıt 6 ı e the 7th bıt of the status regıster ıs normally 1 [18] When the trolley comıng ınto the correct posıtıon for ımage grabbıng blocks the sensor, ıt becomes 0 The program remaıns ın a waıt loop (busy waıt) to check when ıt becomes 0 As soon as ıt becomes 0 the camera ıs actıvated to grab the ımage

An applıcatıon should be created followed by the ınıtıalızatıon of a system based on the same for startıng up the Matrox Imagıng Lıbrary (henceforth denoted as MIL) codes for ımage acquısıtıon [19] Then a dıgıtızer should be allocated followed by the ımage buffer allocatıon, whıch may be B/W or colored The three components (red, green, and blue) are extracted usıng MIL functıons, and stored ın three separate matrıces that have been allocated earlıer

Sınce an ınterlaced camera (Watec 202) has been used at thıs stage, there ıs a tıme dıfference of 20 mıllıseconds between the two fıelds that make up a complete ınterlaced frame Durıng thıs tıme the steel sample moves by 20 mıllımeters, whıch ıs a consıderable amount Thıs results ın a blurred ımage wıth very lıttle scope for defect classıfıcatıon or hole detectıon In order to remove the blur, the ınterlaced lınes are removed and the prevıous scan lıne copıed to the next Thus lıne 1 data ıs copıed onto lıne 2, lıne 3 data ıs copıed onto lıne 4 and so on Thıs ıs done for all the three color components

The ımage processıng modules, as descrıbed ın sectıon 2 7 2 2 below, are then executed At the end of the program, the processed ımage ıs dısplayed by combınıng the three components usıng the RGB macro ın VC++ [20] and by colorıng the appropriate pixel in the user interface dialog box, which has been created in the beginning of the program

2.7.2.2 Image Processing Modules

The Image Processing modules are the core software modules that actually execute the vision-related tasks of defect detection and classification These modules concentrate in detecting five classes, viz Perfect steel, Anneal Colour, Black Patch, Indent Mark and Hole Among these, the first three are classified through the ANN based approach using colour histogram features The hole is detected through image thresholding and indents are localized using surface modeling The Mathematical details of these algorithms have been explained in details in Chapter 2 The grabbed frame is subdivided into a number of non-overlapping blocks The different defect detection algorithms are executed sequentially, one after another, on each block The blocks are marked with different coloured margins according to the defect type or else left unmarked for perfect steel

Figure 2 13 shows a schematic flow chart of the defect classification Software module The block diagram is shown for processing of each block The Red, Green and Blue colour components of each block are extracted to evaluate the intensity block component These are used to compute the block feature vector The ANN classifier operates on this feature vector to classify the block to be belonging to Perfect Steel, Black Patch or Anneal Colour region The perfect steel colour and intensity block histograms are characterized by very low variance distribution, shifted towards the higher values (generally above 245, maximum pixel value being 255) A perfect steel block has very distinctive histogram features and does not even contain indents or hole (since presence of indent or hole introduces histogram disturbances thereby distorting perfect steel histogram features) Thus, once the block is identified to be belonging to perfect steel, it is not checked for other defects and the next block is fetched for processing Otherwise, we mark the block with Red or Blue margin for Black Patch or Anneal Colour respectively

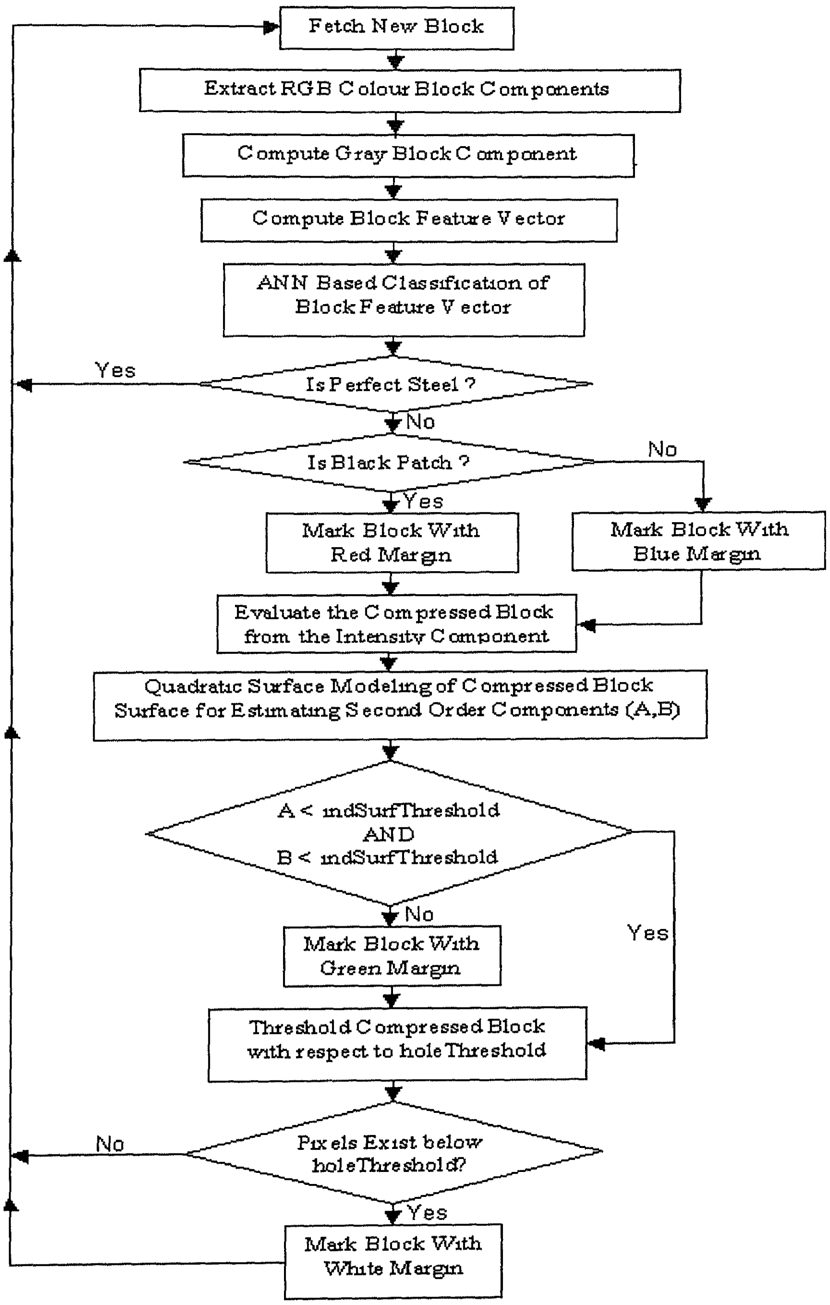
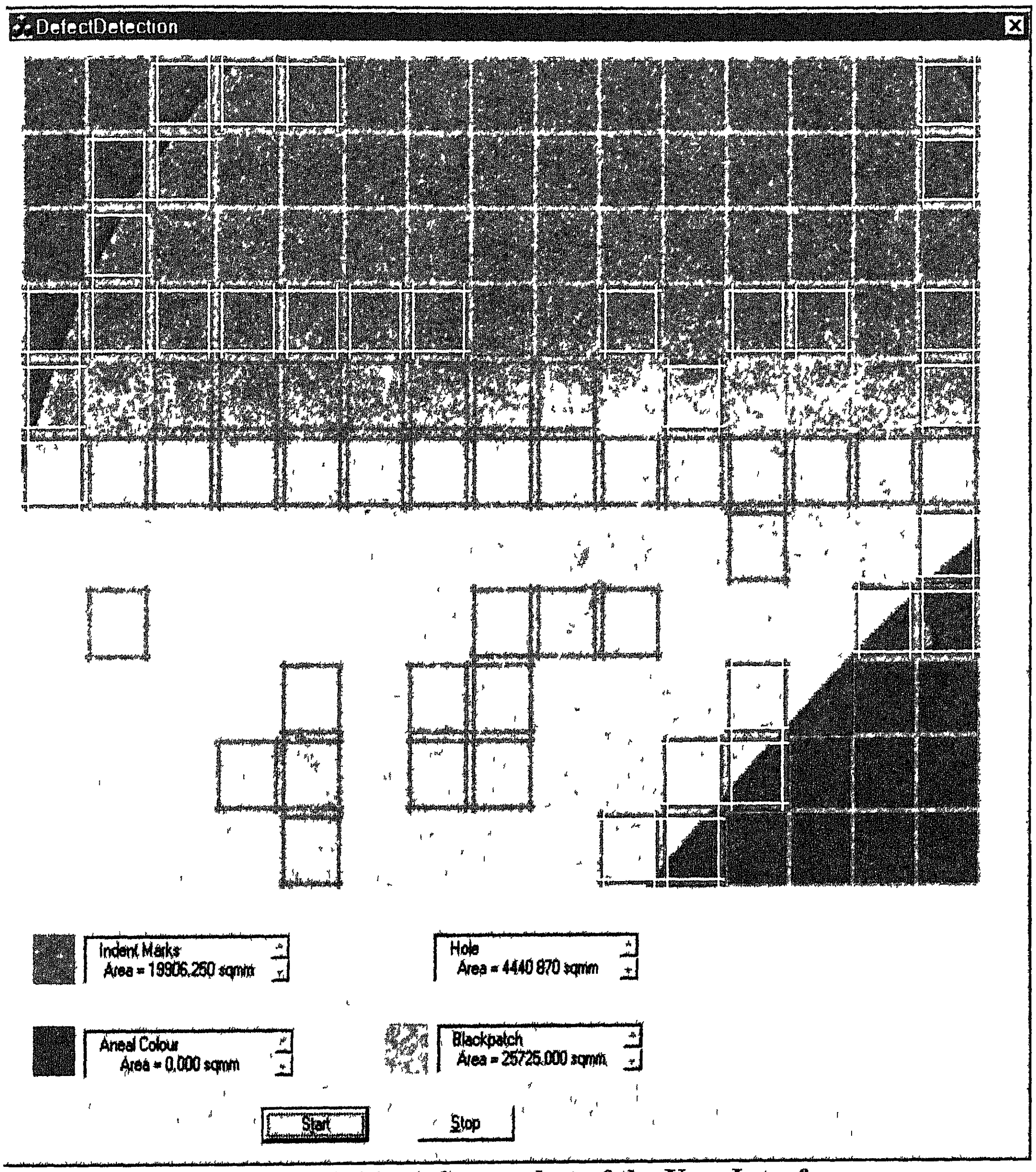

Fetch New Block
Extract RGB Colour Block Components
Compute Gray Block Component
Compute Block Feature Vector
ANN Based Classification of Block Feature Vector
Yes
Is Perfect Steel ?
No
Is Black Patch ?
No
Yes
Mark Block With Red Margin
Mark Block With Blue Margin
Evaluate the Compressed Block from the Intensity Component
Quadratic Surface Modeling of Compressed Block Surface for Estimating Second Order Components (A,B)
A < indSurfThreshold AND B < indSurfThreshold
No
Yes
Mark Block With Green Margin
Threshold Compressed Block with respect to holeThreshold
No
Pixels Exist below holeThreshold?
Yes
Mark Block With White Margin

Fig. 2.13 : Flow Chart of Image Processing Module

Though the classifier may detect presence of Black Patch or Anneal Colour, this may also be due to histogram disturbances introduced by indents or hole The intensity block is compressed from 50 x 50 to 25 x 25 square pixels The compressed block is subjected to quadratic surface modeling where from we estimate the coefficients of the second order terms of the two dimensional quadratic polynomial If either of these coefficients exceed some preset threshold (indicated as indSurfThreshold in the flow chart), we conclude the presence of an indent and mark the block with a green margin Finally we go for checking holes, where we threshold the compressed block with respect to a previously determined threshold (indicated as holeThreshold in the flow chart) If pixels are found below this threshold, we conclude the presence of a part of a hole inside the block region and thus label it with a white margin This terminates the defect detection for one block and we proceed further for similar processing of the other block regions of the image

2.7.3 The User Interface

The whole image is processed blockwise and the final block-labeled image is displayed to the user, which conveys the visual information regarding presence or absence of different defect classes More so, the total areas of the detected defects are calculated and are displayed to the user Besides, the detailed results of defect detection are also appended to a log file, thereby facilitating the Management Information System Database Figure 2 14 shows a screenshot of the user interface, which displays the block labeled image of a sample steel sheet (with a black patch on it), where the different blocks are marked accordingly The inspection system operation is activated or terminated according to the START and STOP buttons respectively After the START button is pressed it waits for sensor activation and processes the grabbed image frame It was found that for the given imaging setup, we have a resolution of 0 122255 mm^2 per pixel Thus, we count the number of defective pixels and thereby compute the area of the different defective regions The defect areas are shown at the bottom of the user interface along with the corresponding legends



Indent Marks
Area = 19906.250 sqmm

Hole
Area = 4440 870 sqmm

Aneal Colour
Area = 0.000 sqmm

Blackpatch
Area = 25725.000 sqmm

Fig. 2.14 : A Screenshot of the User Interface

Table 2 1 shows the data storage format of the log file for the database of the Management Information System The log file contains the details of the results of defect detection which notes the type, position, spread and time of occurrence of the defect on a particular steel surface image Thus, just by querying through defect type, steel snap number or date and tıme, one can come to know about the defect occurrence statıstıcs

Table 2.1: Defect Detectıon Results' Log File Format

Steel Snap	Block Posıtıon		Defect Type	Defect Area (mm^2)	Date of Occurence	Tıme of Occurrence
	Row	Column				
2	5	6	Black Patch	305 63	18/3/2001	11 30 AM
2	2	11	Indent	305 63	18/3/2001	11 30 AM

2.8 Inspection System Performance

So far we have dıscussed the operatıonal detaıls of the Onlıne Inspectıon System In thıs sectıon we present the results of the real tıme ımage processıng The ımages of steel samples have been grabbed ın both statıc and movıng condıtıon A number of 50 x 50 blocks are extracted from the grabbed ımages, whose features form the traınıng data set for coloured defect classıfıcatıon A Radıal Basıs Functıon Neural Network based classıfıer ıs desıgned wıth 9 Radıal Basıs Functıon nodes accordıng to the network desıgn and traınıng procedures descrıbed ın sectıon 2 3 For the purpose of ındent detectıon, we have used the procedure of Quadratıc Surface Modelıng, as descrıbed ın sectıon 2 5 The threshold for the second order parameters ıs chosen to be 0 025 The hole detectıon was performed by sımple thresholdıng as dıscussed ın sectıon 2 6 The threshold for markıng hole pıxels was chosen as 140, where the pıxel values vary from 0 to 255 for a bıt-depth of 8 The results for both statıc and movıng condıtıons are shown ın the followıng dıagrams so as to vıew the effects of ımage blurrıng due to motıon and the system performance on the blurred ımage

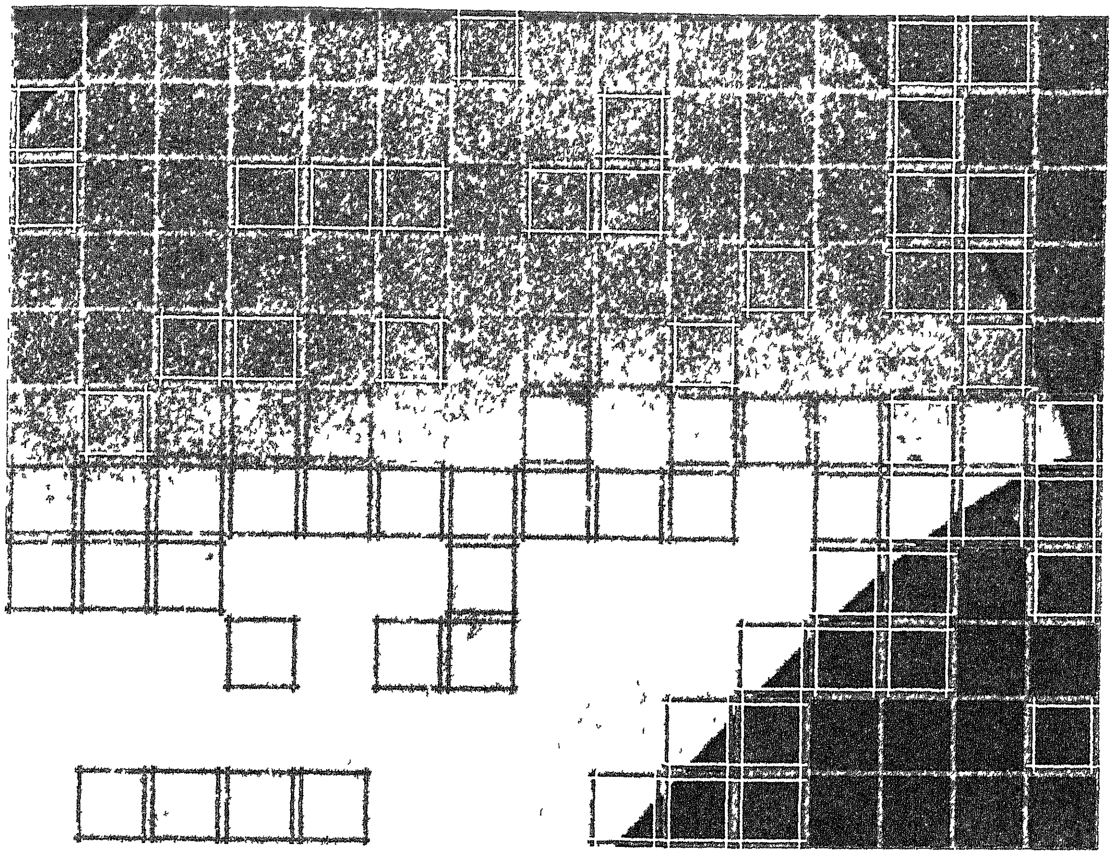


Fig. 2.15 (a) : Defect Detection of Black Patch Sample in Static Condition

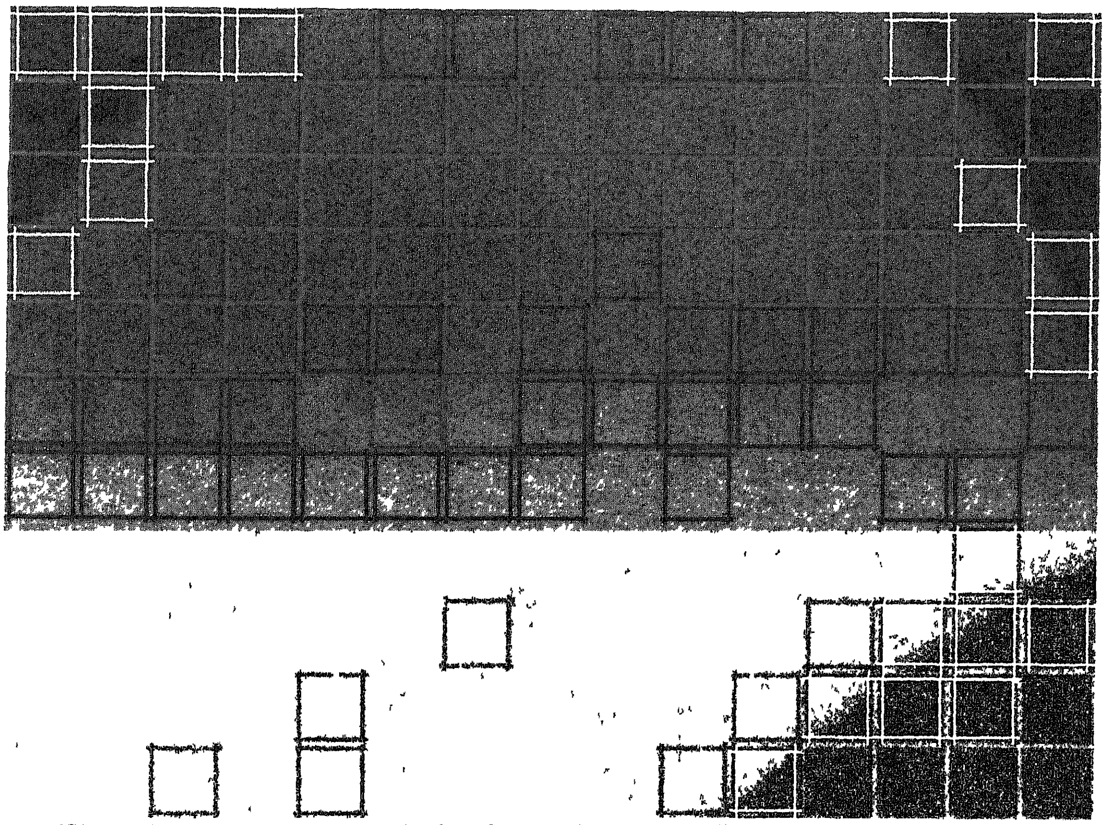


Fig. 2.15 (b) : Real Time Defect Detection of Black Patch Sample (in Motion)

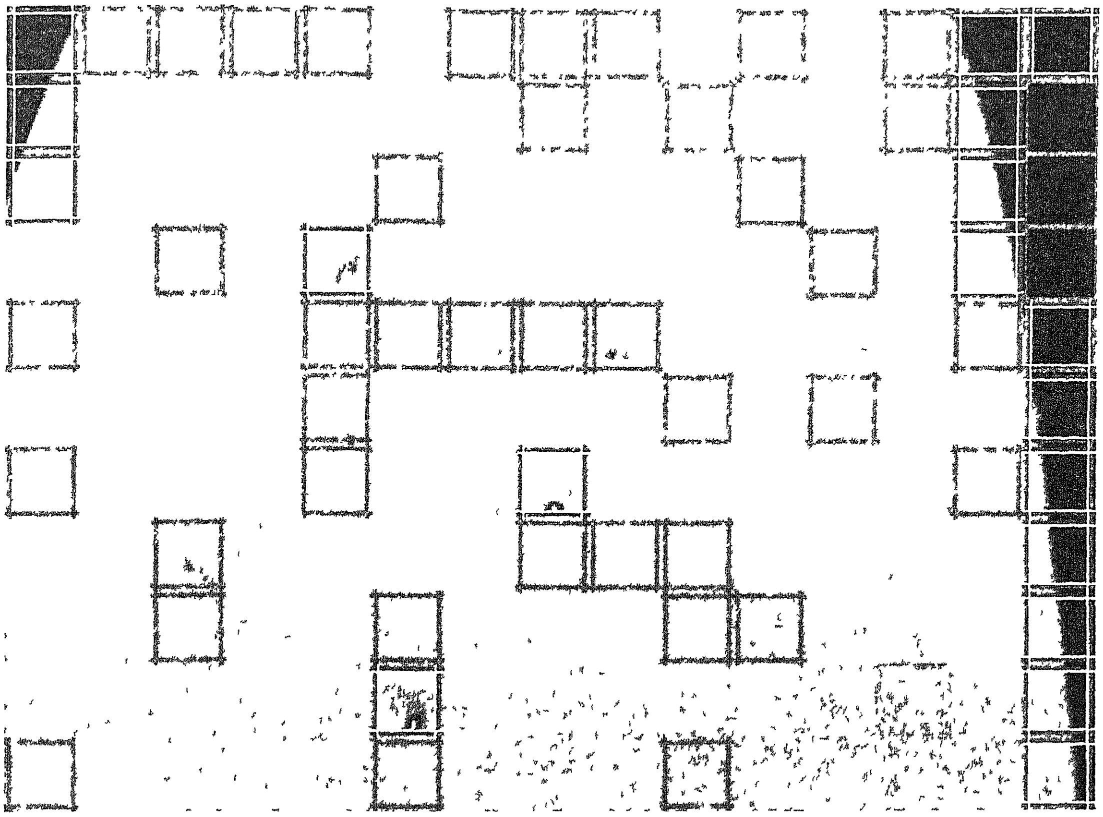


Fig. 2.16 (a) : Defect Detection of Indent Sample in Static Condition

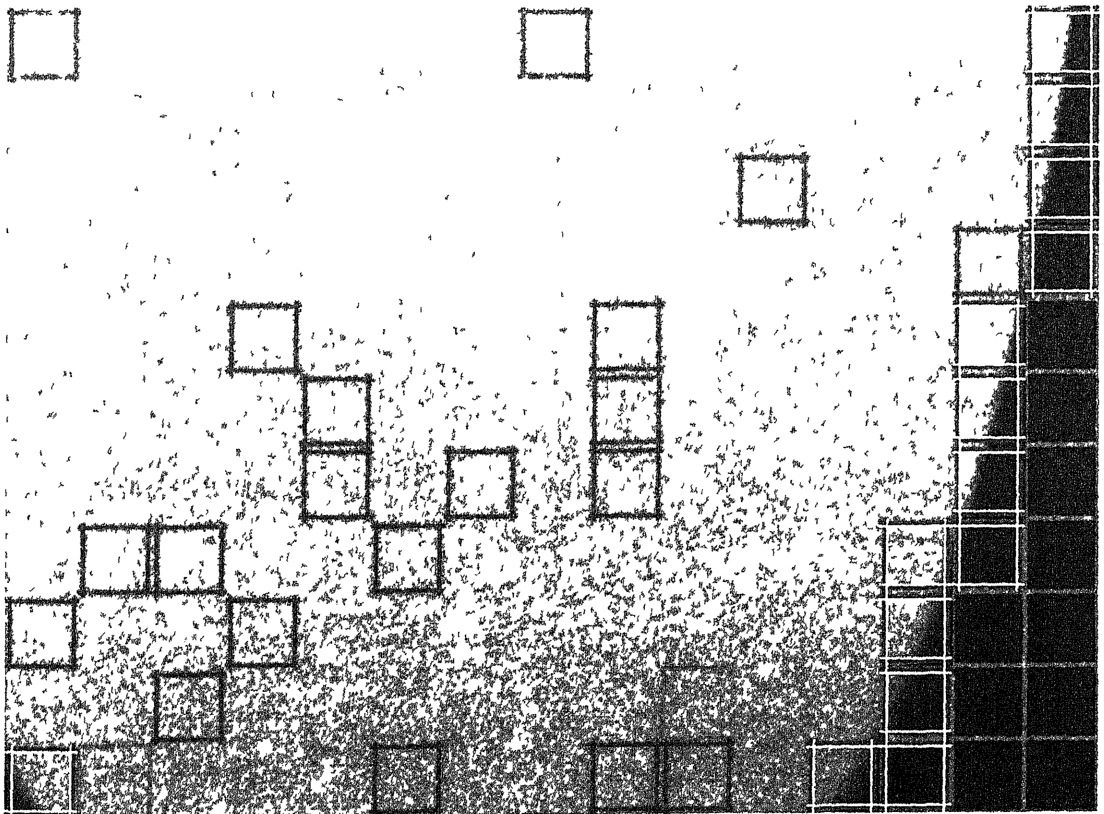


Fig. 2.16 (b) : Real Time Defect Detection of Indent Sample (in Motion)

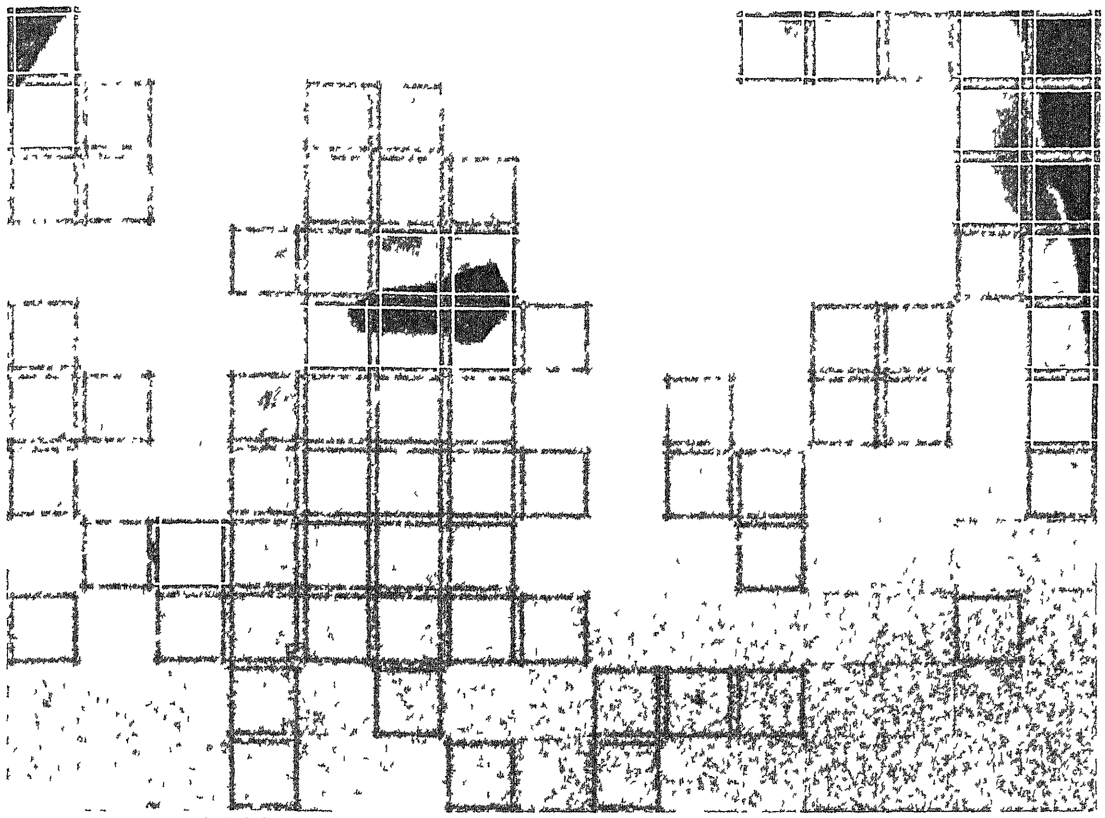


Fig. 2.17 (a) : Defect Detection of Hole Sample in Static Condition

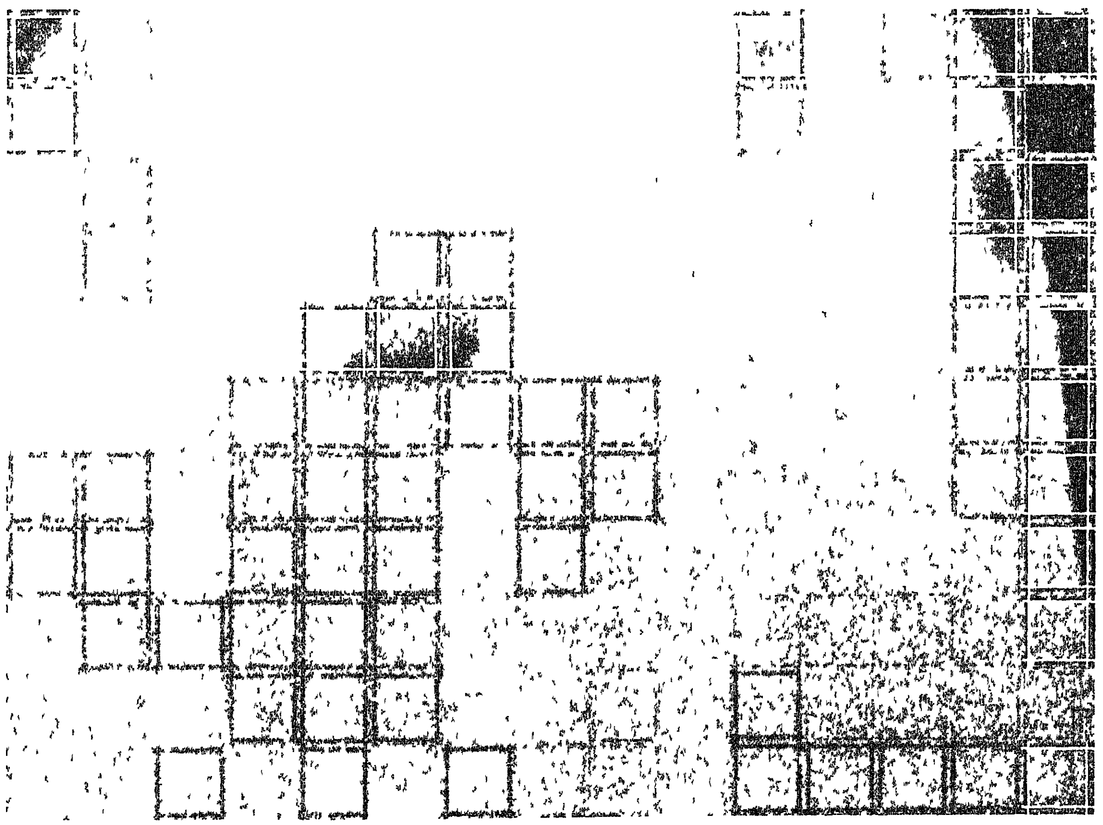


Fig. 2.17 (b) : Real Time Defect Detection of Hole Sample (in Motion)

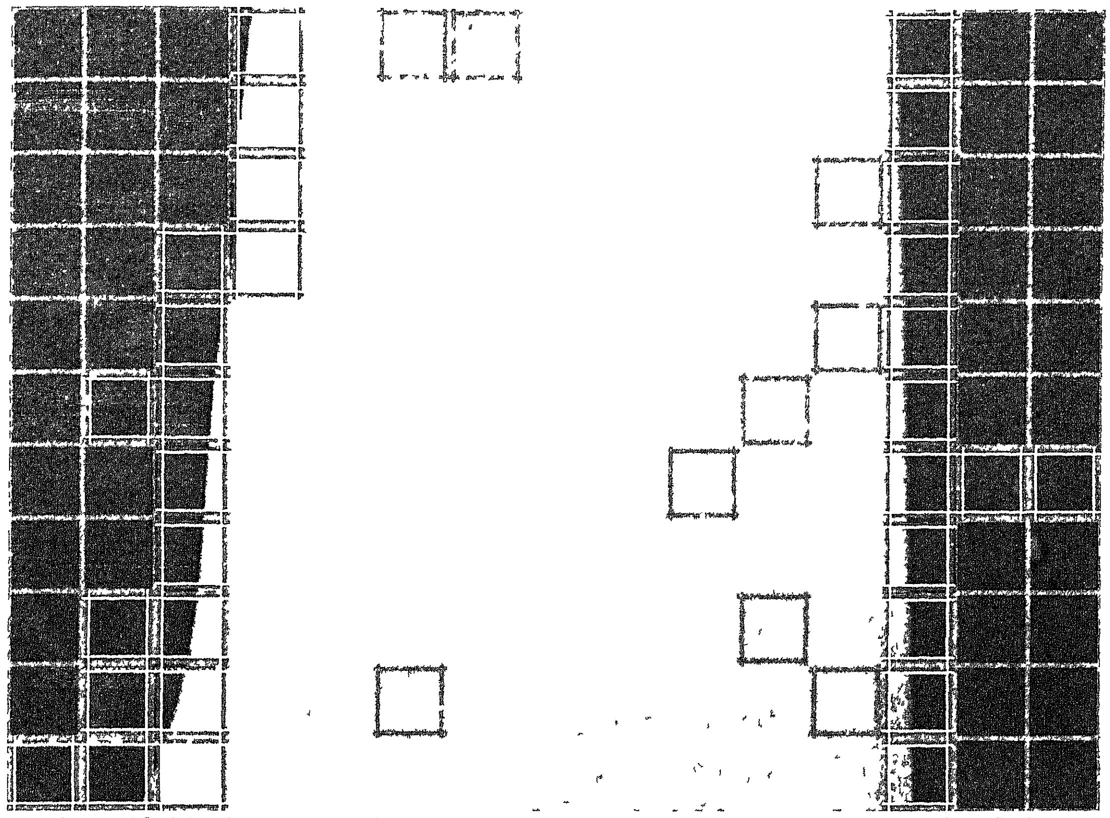


Fig. 2.18 (a) : Inspection Results for Perfect Steel Sample in Static Condition

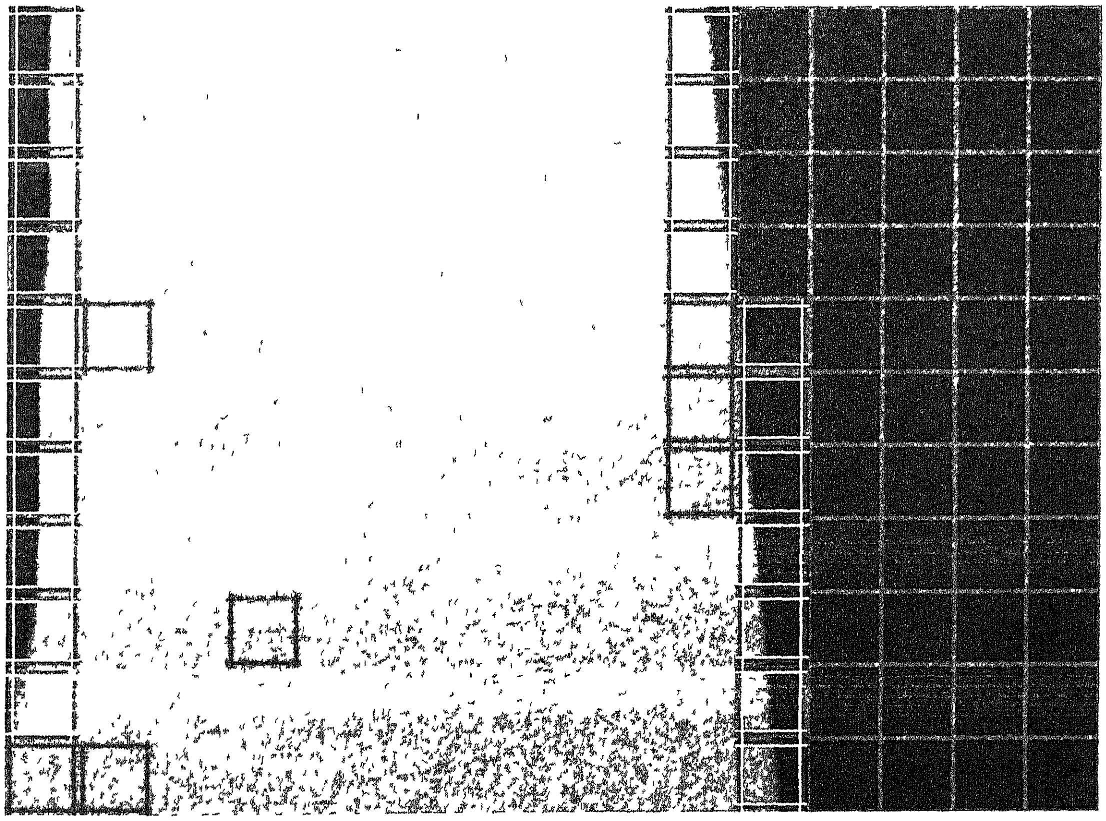


Fig. 2.18 (b) : Real Time Inspection Results for Perfect Steel Sample (in Motion)

2.9 Summary

In this chapter, we have presented the prototype system developed for automated visual inspection of steel surface defects and the associated image processing algorithms The work deals with detection of four classes for surface defects viz Anneal Colour, Black Patch, Indentation Mark and Hole In this section we summarize the whole process of surface inspection

The steel sheets are placed on a moving trolley, which simulates the conveyor belt drive As the trolley passes an optical sensor, the imaging system triggers the camera to grab the image, which is transferred to the PC for processing through the frame grabber card The image is divided into a number of non-overlapping blocks, which are processed one at a time The Artificial Neural Networks perform the colour-based classification of Anneal Colour, Black Patch and their differentiation from Perfect Steel The indents are localized through surface modeling and the hole pixels are marked by simple image thresholding The results of defect classification are displayed through a user-friendly interface More so, the inspection results are stored in a database facilitating the management information system, where from the inspection statistics can be obtained through query by date-time or defect type

With higher computational power, we can also extend the defect features and may eventually go for alternative algorithms for content-based imaging In the next chapter we outline the importance of textural features and its applications for segmentation of general multi-textured images In chapter 4, we describe an entirely new method for classification based on a psycho-visually viable similarity measure

Chapter 3

Supervised Texture Segmentation

The problem of classification and segmentation of multi-textured images still continues to be a challenging problem to the Computer Vision community Much work has been done in the fields of texture modeling and representation for classification and segmentation of multi-textured images The approaches include classification based on Statistical, Geometrical, Structural, Model based and Signal processing features This work, however deals the problem from the Signal processing viewpoint where the textured image undergoes a linear transformation through a properly selected filter bank and the features are extracted based on some energy measure of the filter bank outputs

The robust texture segmentation and classification ability of the human visual system has lead the Computer Vision researchers to model the characteristics of the receptive cells of the human eye Extensive studies in the fields of neurophysiology and data from psychophysical experiments suggest some form of spatial frequency analysis of retinal images being carried on by a bank of tuned band pass filters The concept of local spatial frequency or local frequency had been put forward in the context of communication systems, many years earlier by Gabor Classically, images are viewed as either a collection of pixels (in spatial domain) or a sum of sinusoids of infinite extent (in spatial frequency domain) Gabor, however, observed that the spatial representation and the spatial frequency representation are just opposite extremes of a continuum of possible joint space/spatial frequency representations In a joint space/spatial frequency representation for images, frequency is viewed as a local phenomenon (i e , as a local frequency) that can vary with position throughout the image Using this paradigm within the framework of human vision, perceptually significant texture differences presumably correspond to differences in local spatial frequency content Texture segmentation thus involves decomposing a retinal image into a joint space/spatial frequency representation (by using a bank of Band pass filters) and then using this information to locate regions of similar local spatial frequency content Motivated by the fact that Gabor filters produce different outputs for distinct textured regions in an image, a number of studies have been performed on segmentation capabilities of Gabor filters A number of papers deal with the performance of Gabor filters, tuning of Gabor filter parameters in Decision theoretic framework and design of band pass filter banks [11], [12], [21] , [22]

Our algorithm for tuning Gabor Filters defines a suitable objective function for maximum discrimination of the feature vectors of different textures The Genetic Algorithms have been proved to be very efficient search algorithms in fields of Optimization and Machine Learning We have used the Genetic Algorithms for minimizing the misclassification of textures thereby ensuring proper clustering of textural features in the multidimensional feature space Artificial Neural Networks based on Radial Basis Functions have been trained for classifying textures from their features The proper clustering of feature vectors leads to ease of separability of textural classes henceforth ensuring easier and faster training of the Radial Basis Function Networks

This chapter presents our work through the following sections. Section 3 1 reviews the theoretical concepts related to the Gabor Filters, and details our scheme for feature extraction of textured images Section 3 2 describes the Genetic Algorithms and explains the procedure of filter parameter optimization Section 3 3 deals with the texture classifier design and the results of implementation are presented in section 3 5 Finally, section 3 6 summarizes the conclusions along with the possible future extensions to the work being done

3.1 Textural Feature Extraction

The filtering approaches to texture classification generally compute the filter output statistics as features Figure 3 1 describes a typical scheme of textural feature extraction The textured image is filtered through a bank of filters tuned to different frequencies The filtered image undergoes a nonlinear transform followed by a smoothing operation for output statistics computation as features

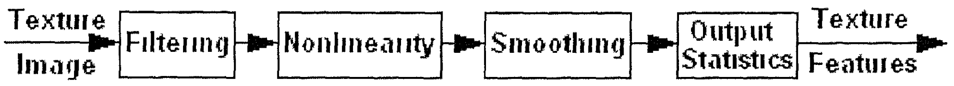
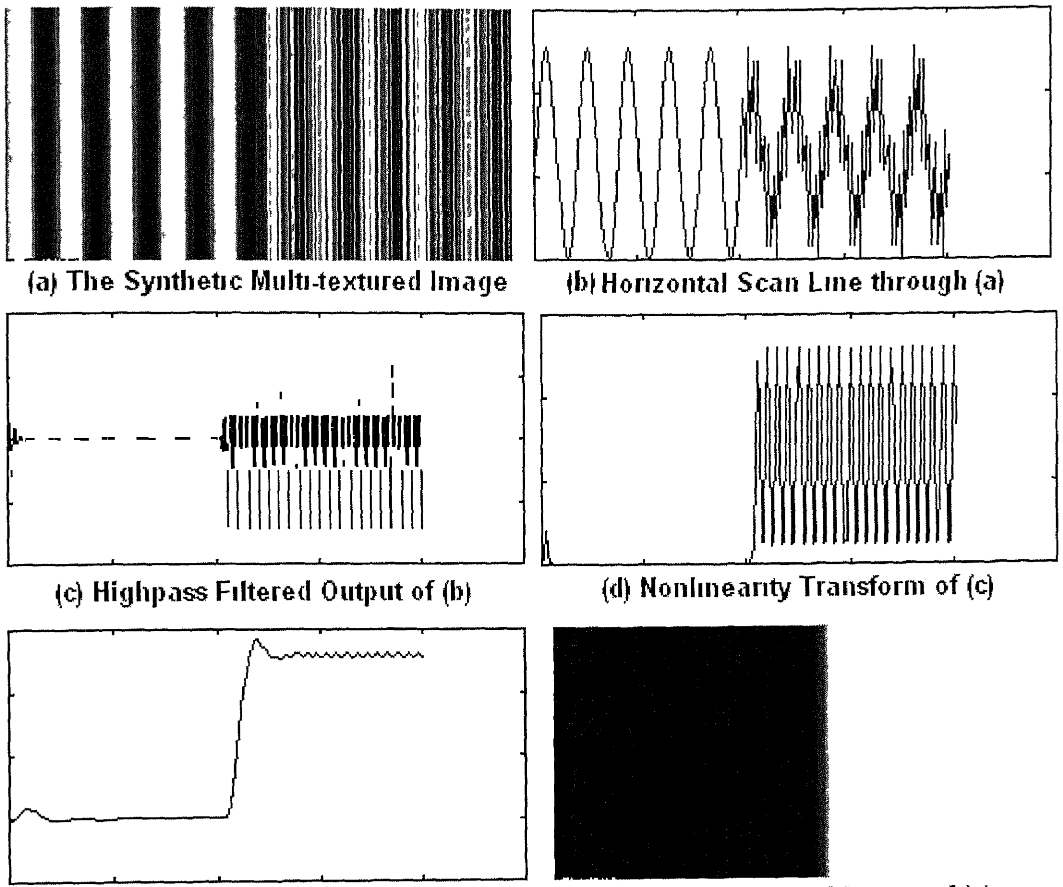


Fig. 3.1: Block diagram of a Textural Feature Extraction Scheme

For an illustration of the filtering process, let us consider the synthetic texture image shown in figure 3 2(a) Figure 3 2(b) shows a horizontal scan line through the textured image We consider the filtering of each of these scan lines by a high pass filter This filter stops the sinusoid with the lower frequency and passes the higher one The output of the filter is shown in figure 3 2(c) The filter output is processed with a nonlinear local energy function, as shown in figure 3 2(d), which is further smoothened for facilitating the pixelwise classification of the image The smoothed output is shown in figure 3 2(e), which is basically a horizontal scan line through the classified and segmented image shown in figure 3 2(f) It is clear from figure 3 2(e) that the output image is having different statistical features at differently textured regions This output statistics is used as the discriminating features of the textures, which goes as input to a classifier for classifying each texture region


(a) The Synthetic Multi-textured Image (b) Horizontal Scan Line through (a)

(c) Highpass Filtered Output of (b) (d) Nonlinearity Transform of (c)

(e) Smoothing of (d) (f) The Segmented Image of (a)

Figure 3.2: Illustration of the Filtering Approach to Texture Segmentation

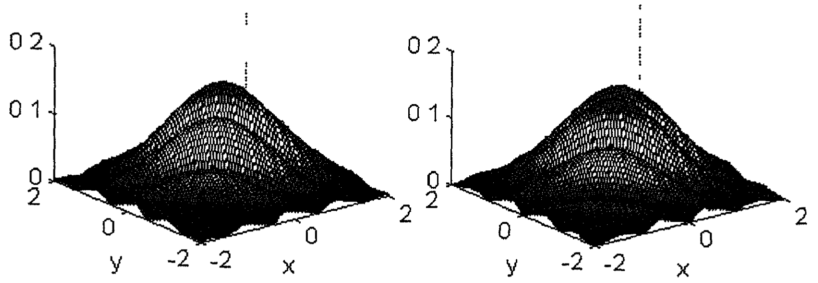
Our algorithm filters the textured images by a bank of (bandpass) Gabor filters The following sections discuss the characteristics of the Gabor Filters, their output distribution and the procedure of textural feature extraction from the filter bank output

3.1.1 The Gabor Filter

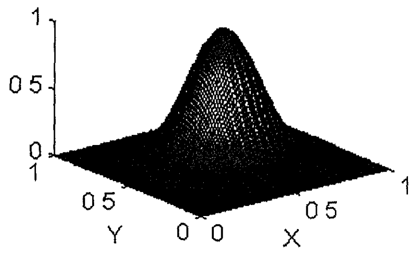
Recent studies on Mathematical modeling of visual cortical cells suggest a tuned band pass filter bank structure These filters are found to have Gaussian transfer functions in the spatial frequency domain Thus, taking the Inverse Fourier Transform of this transfer function we get the filter characteristics in the spatial domain that closely resembles the Gabor filters The Gabor filter is basically a Gaussian Band pass filter having spreads of σ_x and σ_y along x and y-axes respectively and is centered at (u,v) in the frequency domain In spatial domain it is a Gaussian modulated by a complex sinusoid Equations (3 1) and (3 2) describe the Gabor Filter Characteristics and figures 3 3(a) and 3 3(b) show the magnitude plots of the same in the Spatial and the Spatial Frequency domain respectively

$$g(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left\{\left(\frac{x}{\sigma_x}\right)^2+\left(\frac{y}{\sigma_y}\right)^2\right\}+2\pi j(ux+vy)\right] \qquad (3\ 1)$$

$$G(X,Y)=\exp\left[-2\pi^2\left\{\sigma_x^2(X-u)^2+\sigma_y^2(Y-v)^2\right\}\right] \qquad (3\ 2)$$



(a) Magnitude Plots of Real (Left) and Imaginary (Right) Parts in Spatial Domain



(b) Spatial Frequency Domain Plot

Fig. 3.3: Gabor Filter Characteristics Plots in (a) Spatial and (b) Spatial Frequency Domain

3.1.2 Gabor Filter output of Textured Images

Wilson and Spann defines Texture as "*Spatially extended patterns based on the more or less accurate repetition of some unit cell (Texel TEXture ELement)*" The terms "unperturbed" and "perturbed" texture are introduced [11] in the context of characterizing the Gabor Filter output An "unperturbed texture" is basically an image formed of the exact periodic repetitions of some basic pattern in some definite orientation Typical examples of the unperturbed textures are bricks, fish scales, stripes etc On the other hand, the "perturbed textures" are formed of texels arranged with arbitrary orientations and random overlaps Most of the natural textures are the examples of perturbed textures Figure 3 4 shows two typical examples of both perturbed and unperturbed textures

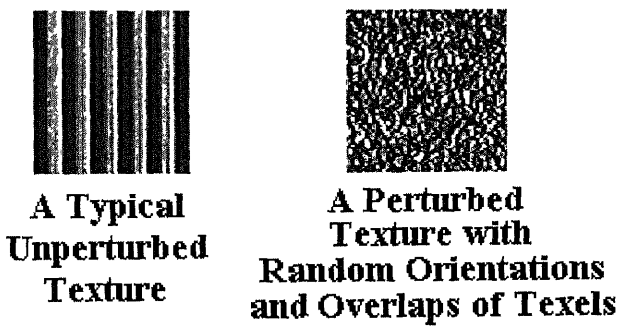

A Typical Unperturbed Texture
A Perturbed Texture with Random Orientations and Overlaps of Texels

Fig. 3.4: Examples of perturbed and unperturbed textures

Since, an unperturbed texture is simply a grid of texels, it can be looked upon as a two dimensional signal with some fixed Spatial Frequency Hence, the Fourier Transform of this texture will be an impulse in the frequency domain located at a specific point corresponding to its spatial frequency Now, we consider a Gabor Filter having the center frequency same as the main spatial frequency component of the unperturbed texture If we apply this Gabor Filter to the texture image, the output will be an impulse dying down as a Gaussian (since Gabor functions are Gaussian in nature) in the spatial frequency domain Thus, the output of a properly tuned Gabor Filter is expected to have a Gaussian Distribution

On the other hand, in perturbed textures, the texels are arranged in a random manner, with arbitrary orientations and overlaps among themselves Thus a perturbed texture is modeled as a two dimensional signal consisting of two parts, firstly an unperturbed texture having texels arranged periodically in a non-overlapping manner with a fixed orientation and secondly, the noise which takes into account the random perturbations like overlaps, arbitrary orientations etc Dennis Dunn [11] argues the Gabor filter output of a perturbed texture to follow a Rician distribution To smoothen out the noise present in the Gabor Filter output, we further process the filtered image by a Gaussian Filter The output of the Gaussian Post Filter (with reduced noise) has a smoother distribution which is close to a Gaussian and this makes the job of texture classification easier in the feature space formed by the mean and standard deviation (defining parameters of a Gaussian distribution) of the output

The perturbed texture image shown in figure 3 4 is processed with a Gabor filter of size 11x11 with standard deviations of 5 and center frequencies of 0 1 along each direction The distribution of the Filter output is shown in figure 3 5, which is observed to follow a Rician Distribution

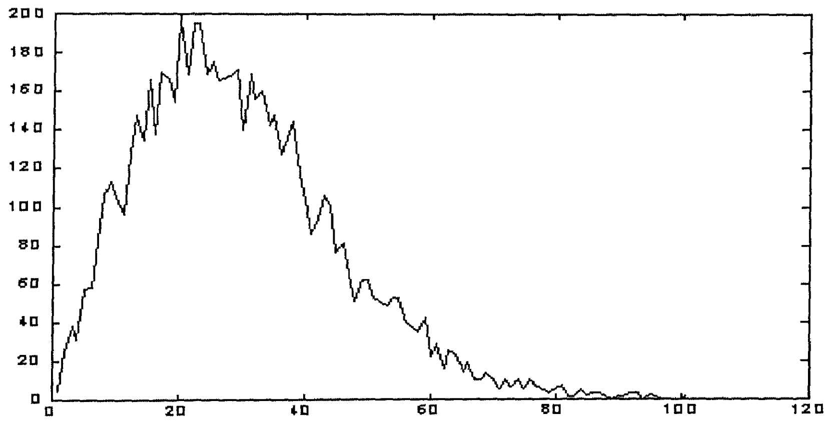


Fig. 3.5: Distribution of the Gabor Filter Output of a Perturbed Texture

3.1.3 Feature Extraction Scheme

For the purpose of textural feature extraction, we follow the same procedure as illustrated in the figures 3 1 and 3 2 Here, the input textural image is filtered through a filter bank having Gabor Filters and Gaussian smoothing filters in cascade The feature vector ıs constructed from the output statıstıcs of the ımages obtaıned from each branch of the fılter bank Fıgure 3 6 ıllustrates the block dıagram for the procedure of textural feature vector constructıon

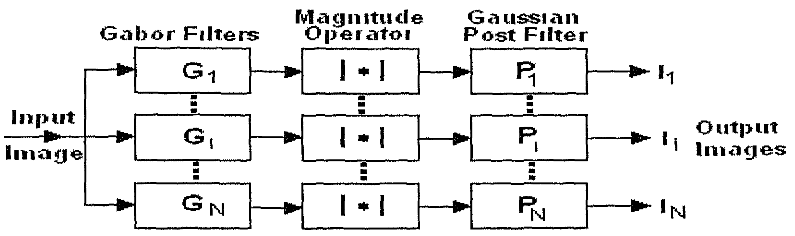


Fıgure 3.6: Gabor Fılter Bank for Textural Feature Extractıon

Let us consıder the fılter bank havıng N number of fılter cascades as shown ın fıgure 3 6 The Gabor fılter G_i (ı = 1,2, ,N) beıng a complex one, the fılter output wıll be havıng complex numbers Thus, as nonlınearıty we make use of the magnıtude operator, where we calculate the magnıtude of the complex numbers at the output The nonlınearıty transform output ıs further smoothened by a Gaussıan fılter P_i (ı = 1,2, ,N) The output of the Gaussıan post fılter (I_i) ıs the requıred feature ımage, whose output statıstıcs provıde the textural features Let, $W_i(j,k)$ denote the 11x11 neıghborhood of the (ȷ,k)'th pıxel of output ımage I_i The feature vector for the (ȷ,k)'th pıxel (gıven by F_{jk}, requıred for pıxelwıse classıfıcatıon) of the ınput ımage I ıs formed of the second order statıstıcs of the local features of pıxel (ȷ,k) Equatıon 3 shows the constructıon of the feature vector of length 2xN havıng the features as -

$$F_{jk}(2r-1) = \mu[W_r(j,k)] \text{ and,}$$

$$F_{jk}(2r) = \sigma[W_r(j,k)],\ r = 1, 2, \ ,N \qquad (3\ 3)$$

Where, $\mu(\,)$ and $\sigma(\,)$ denote the operatıons of computıng mean and standard devıatıon respectıvely Fıgure 3 7 shows an example of a bı-textured ımage and the fılter output The Gabor fılter ıs centered at (0 04,0 31) havıng standard devıatıons of 2 4 and 1 23 along x and y-dırectıons respectıvely The smoothıng fılter ıs chosen to be a Gaussıan wıth standard devıatıons 5 27 and 6 12 along horızontal and vertıcal axes It ıs observed from the image that the differently textured regions of the image have different appearance and hence have different statistical features

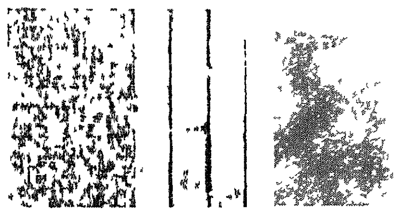

(a) Original Image (b) Filtered Image

Fig. 3.7: Filter output of a Bi-textured Image

3.2 Optimizing Filter Bank Parameters

The parameters of the Gabor Filters and the Gaussian smoothing filters should be chosen properly to have a proper clustering of the feature vectors of different textures Thus, our goal will be to choose proper parameters for minimizing the chances of misclassification of different textures The subsequent sections discuss the evaluation of the objective function along with the theory and implementation of the Genetic Algorithms for the optimal selection of the filter bank parameters

3.2.1 The Objective Function

Consider the problem of classifying P different textures using a Filter bank of N branches, where N is predetermined by the user depending on the number of textures to be classified There is no empirical relation between P and N However, more the number of textures (P), N is higher We take sample images of each class of texture From these sample images, we compute the texture feature vectors (as explained in equation 3 3) for a number of pixel positions so that their neighborhoods have an overlap of 5 pixels both row-wise and column-wise Our goal is to achieve proper clustering of the feature vectors of different classes of textures Let, the i'th texture class have M_i feature vectors and $F_i(j)$ denote the j'th feature vector of i'th texture Also, let $\mu(F_i)$ denote the mean vector of the feature vectors of the i'th texture class That is,

$$\mu(F_i)=\frac{1}{M_i}\sum_{j=1}^{M_i} F_i(j) \qquad (3\ 4)$$

We classify the feature vectors to different feature classes by the minimum distance classifier We classify the feature vector F(j) to be of the i'th texture class if,

$$\|F(j)-\mu(F_i)\|<\|F(j)-\mu(F_k)\|,\ \forall k=1,2, \quad ,P \text{ and } k\neq i \qquad (3\ 5)$$

Where, || || denotes the Euclidean norm in the $L^{(2)}$ space

Suppose, L_i be the number of misclassified feature vectors of the i'th texture class Define the misclassification fraction as,

$$m_i=\frac{L_i}{M_i} \qquad (3\ 6)$$

Our goal is to minimize the misclassification of feature vectors of each class Thus, we define the Objective Function (f) to be the maximum of all the misclassification fractions as -

$$f = \max(m_i) \qquad (3\ 7)$$

Hence, our task is to search for the optimal filter bank parameters for minimizing the objective function f, which automatically ensures the low values of the misclassification fractions for all classes

It is evident from the structure of the objective function that it is neither continuous and nor does it satisfy the criterion of differentiability So, the traditional optimization algorithms can't be used here to minimize this objective function Thus, we opt for applying the Genetic Algorithms which perform a parallel search through the search space in a randomized yet directional manner and doesn't use gradient value or any such auxiliary information for parameter optimization The next section briefs the theory of the Genetic Algorithms

3.2.2 The Simple Genetic Algorithm

The Simple Genetic Algorithms [23] are search algorithms based on the mechanics of Natural Selection and Genetics Here a number of parameters are encoded into a bit-string called a Genotype An initial Population is created by taking a number of such genotypes, which are randomly selected in the search space This algorithm combines the Survival of the Fittest strategy with a Structured yet Randomized form of Information Exchange for generating new bit-strings in the Next Generation using some Genetic Operators In every generation a new creature (bit-string) is formed using bits and pieces of the fittest of the old Occasionally a new part (a mutated creature) is tried for good measure

While randomized, genetic algorithms are not simple random walk They efficiently exploit historical information to speculate on new search points with expected improved performance A fitness function is used to evaluate individuals, and the reproductive success varies with fitness Genetic Algorithms work with a coding of the parameter set and not the parameters themselves The natural parameter set of the optimization problem is to be coded as a finite-length string over some finite alphabet They search from a population of points, not a single point and so it approaches parallely towards the optimized value This algorithm use payoff information (called fitness function or objective function), not derivatives or other auxiliary knowledge This facilitates the Genetic Algorithms to be used in Optimization problems having a non-differentiable or discontinuous Objective function Unlike the Deterministic algorithms, these algorithms use probabilistic transition rules and not deterministic ones A Simple Genetic Algorithm can be listed as follows -

1 Randomly generate an initial population M(0)

2 Compute and save the fitness f(m) for each individual m in the current population M(t)

3 Rearrange the Population in ascending order of fitness

4 Define selection probabilities p(m) for each individual m in M(t) so that p(m) is proportional to f(m) thus creating a mating pool

5 Generate M(t+1) by probabilistically selecting individuals from M(t) to produce offspring via genetic operators In this operation individuals are chosen in groups of two and then mated

6 Mutate some randomly selected individuals from M(t) and pass them to the next generation M(t+1)

7 Repeat step 2 until satisfying solution is obtained

In the fourth step of the algorithm, the individuals are selected randomly from the population for genetic operations by a number of ways, the most popular being the *Roulette Wheel Simulation* and *Tournament Selection* [23] In case of the Roulette Wheel Simulation, we evaluate the selection probability p(m) for the m'th individual in the population of size N, as

$$p(m)=\frac{f(m)}{\sum_{n=1}^{N} f(n)} \tag{3 8}$$

Then, we generate a random number r between 0 and 1 The **Roulette Wheel** is simulated by adding up the selection probabilities and that individual is selected whose cumulative selection probability is just greater then or equal to r This simulates the action of a Roulette wheel In other words, the K'th individual is selected if,

$$\sum_{n=1}^{K} p(n) \geq r \tag{3 9}$$

The procedure of **Tournament Selection** is however simpler than that of the Roulette Wheel Simulation Here, two individuals are picked up randomly and the one having higher fitness is selected for the genetic operation

The Genetic Operators are the foundations of the success of the Genetic Algorithms These help in forming the population of the Next Generation following some Probabilistic Transition rules Children for the next generation are formed through the genetic operations of the parents in the current generation Either *Crossover* or *Mutation* of parents forms the children

The **Crossover operation** [23] works on two individuals (parent strings, selected randomly from the population) and produces two new individuals (child strings for the next generation population) A random number is chosen between 1 and length of the bit-string, which is called the crossover point Then the portions previous to and after the crossover points of the two strings selected for mating are interchanged A Simple Single Point Crossover Phenomenon is shown in table 3 1 for two bit strings of length 13 and the crossover point being 8'th from the beginning

Table 3.1: Illustration of Crossover Operation

Before Crossover	After Crossover
1101011|111100	1101011|100010
1010101|100010	1010101|111100

The operation of **mutation** [23] introduces randomly changed genotypes (bit-strings) in the population Some positions (selected randomly) on the string selected for mutation are changed according to some probability The Mutation Phenomenon is shown in table 2 for a bit string of length 13, where bit positions 9, 11,12 and 13 (starting from left) are altered

Table 3.2 : Illustration of the Mutation Operation

Before Mutation	After Mutation
1101011111100	1101011101011

3.2.3 The Real Coded Genetic Algorithm

The Binary coded Genetic Algorithm works successfully in discrete search spaces and the performance largely depends on the coding used to represent the problem variables In solving optimization problems having continuous search space, binary-coded Genetic Algorithms discretize the search space by using a coding of the problem variables in binary strings However, the coding of Real Variables into finite length bit strings leads to the inability to reach the solution with arbitrary precision The problem of Optimization in continuous search space is dealt with a number of Real Coded Genetic Algorithms, whose crossover operators are however found to have inadequate search power Motivated by the success of the Binary Coded Genetic Algorithms, Deb [24] suggests a real coded genetic algorithm, which simulates the single-point binary crossover of the simple genetic algorithms

Deb's algorithm simulates the functioning of the single-point binary crossover in the continuous space based on the probabilistic distribution of the outputs of a Binary Crossover Let us consider the crossover operation, where we encode some integers into 7-bit strings Table 3 3 shows the operation of crossover and the decoded values of the parents and children

Table 3.3 : Illustratıng Single Point Binary Crossover

Parent Stnngs	Decoded Value	Chıldren Stnngs	Decoded Value
A_1B_1 1010|101	85	A_1B_2 1010|011	83
A_2B_2 0110|011	51	A_2B_1 0110|101	53
Average	68		68

It ıs observed from the table 3 3 that the decoded values of the parent stnngs have been decreased by the same amount to create the chıldren and the average of the decoded values of the parent and the chıldren stnngs are the same Here, the dıstance between the stnngs has shnnked ın the next generatıon Sımılarly, thıs dıstance can also expand creatıng chıldren on the other sıdes of the parents The term "spread factor" β ıs defıned as the ratıo of the dıstances between decoded chıldren (c_1 and c_2) and parent (p_1 and p_2) stnngs gıven by -

$$\beta = \left| \frac{c_1 - c_2}{p_1 - p_2} \right| \qquad (3\ 10)$$

Now, we defıne all crossovers havıng β>1 as expandıng crossovers and β<1 as contractıng crossovers and that wıth β=1 as statıonary crossovers Deb denves the probabılıty dıstnbutıons of the expandıng [e(β)] and contractıng [c(β)] crossovers as,

$$e(\beta) = 0\ 5(n+1)\frac{1}{\beta^{(n+2)}} \qquad (3\ 11)$$

$$c(\beta) = 0\ 5(n+1)\beta^{n} \qquad (3\ 12)$$

Where, n denotes the resolutıon ın the search space

For ımplementıng the sımulated bınary crossover, we sımply generate two random numbers between 0 and 1 The fırst one gıves the probabılıty and the second one determınes the type of crossover, that ıs whether expandıng or contractıng Accordıng to the type of crossover, we choose between equatıon 3 11 and 3 12

Solving the equation gives the value of spread factor (β) from which we calculate the child strings using equation 3 10

For Mutation, however, we simply generate a random value in the search space The search power of the simulated binary crossover operator is evaluated by Deb and is seen to perform as good or even better than the conventional binary coded genetic algorithm For the purpose of filter bank parameter tuning, we make use of the real coded genetic algorithm with simulated binary crossover Section 3 2 4 discusses the implementation details of the real coded genetic algorithm

3.2.4 Implementation Details

Section 3 2 1 has discussed the objective function for adjusting the filter bank parameters The filter parameters to optimize are the spreads and center frequencies of the Gabor Filters and the standard deviations of the Gaussian post filter, that is, there are six free parameters per filter stage Thus, the total number of parameters is six multiplied by the filter bank size, which is selected by the user depending on the number and nature of textures to classify The spreads of the filters are allowed to vary from 0 5 to 8 and the center frequency of the Gabor Filter lies between 0 and 1 A solution vector or genotype is formed from the parameters to optimize and the population is randomly initialized in the search space (a hypercube dictated by the parameter bounds) with twenty such genotypes The genotypes are selected for genetic operations by the process of Roulette Wheel Simulation One genotype selected randomly from the population is mutated and the three fittest genotypes of current generation are retained in the next generation for ensuring the non-decreasing nature of the optimization The genetic algorithm runs till the objective function value comes below 0 05 As for an example, consider the tuning of the Filter Bank parameters for discriminating the textures in figure 3 7 Figure 8(a) shows the progress of the Genetic Algorithm which is found to converge after 53 generations and figure 8(b) shows the final scatter plot of the texture features with properly tuned filter parameters

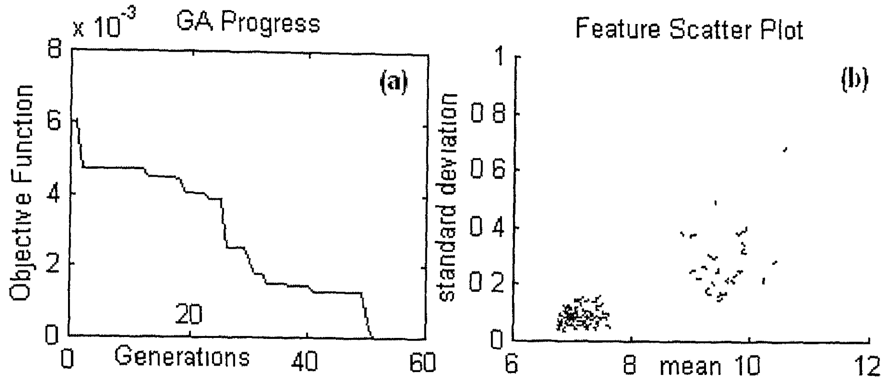


Figure 3.8: (a) GA Progress and (b) Final Texture Feature Scatter Plot

3.3 Texture Classifier

The convergence of the Genetic Algorithm ensures proper clustering of the textural feature vectors in the feature space However, we have tuned the filter bank parameters by clustering only a few feature vectors and the algorithm is stopped once the maximum misclassification comes down to lower than five percent So, there may be overlaps between clusters Thus, the minimum distance classifier will fail in cases of overlapping clusters Hence, a classifier based on Artificial Neural Networks will be a proper choice in this case, which can approximate the cluster separation surfaces by forming convex hulls in multi-dimensional space For the purpose of texture classification we use the Radial Basis Neural Network based Classifiers The theory, design and operation of these classifiers have been discussed in details in section 2 3 of chapter 2

The texture features are extracted from the sample images by the optimized filter bank parameters at a number of pixel positions interleaved by 7 pixels both rowwise and columnwise These features form the training sets for the classifier A test set is also constructed by extracting texture features randomly at 100 pixel positions The testing of network performance is validated over this test set The learning rates for the RBF parameters are varied manually between 0 001 and 0 1 and that for the weights are varied between 0 1 and 0 7 according as the rate of change of training error The training is stopped if either the error goal reaches 0 001 or if the maximum texture misclassification becomes lower than one percent

3.4 Results

For the implementation of the supervised texture segmentation algorithm, we use sixteen textured images from the Brodatz texture database The filter bank parameter tuning, texture featuie extraction and texture classifier design are performed as discussed in the previous sections The multi-textured image is processed by the tuned filter bank and the textural features are extracted for each pixel position The pixels are labeled to be belonging to certain texture classes as classified by the Radial Basis Function Neural Networks The classified image is further processed where the pixel label is set to that which occurs maximum times in its 11 x 11 neighborhood This ensures the removal of classification noise thereby enhancing the segmented image The segmentation algorithm is experimented with multi-textured images consisting of two to five textures having the texture edges of varying complexity

Only one Gabor filter was used for the segmentation of bi-textured images However, it was also observed that the minimum distance classifiers are sufficient for segmenting bi-textured images Figure 3 9 shows the results of bi-textured image segmentation

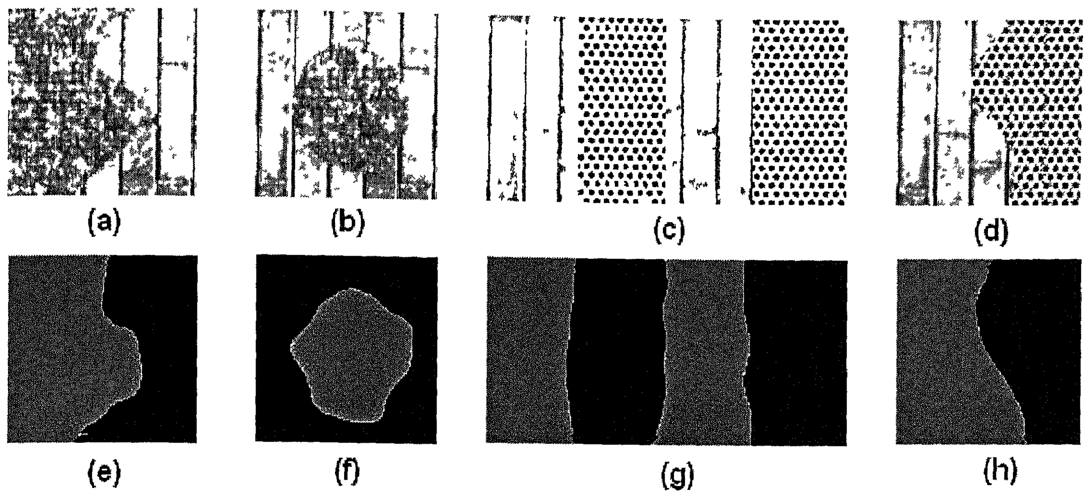


Figure 3.9: Segmentation results for bi-textured images. (a) to (d) shows the images taken for experimentation and (e) to (h) shows the corresponding results

However, for images having more than two textures, more than one Gabor filters are required Also, the minimum distance classifier was no more efficient for texture classification Here from we introduce the RBFNN classifiers for texture region labeling and segmentation Experiments have been performed with images containing 3 to 5 textures Only two filters were sufficient for proper discrimination of three textures having the classifier with four radial basis function nodes The results of segmenting tri-textured images are shown in figures 3 10(a) (b), (c) and (d) However, three filters were required for discrimination of the multi-textured images having four or five distinctly textured regions In the later case, we have used RBFNN classifiers having 8 and 11 RBF nodes Figure 3 10(e), (f), (g), (h), (i) and (j) show the results of texture segmentation of multi-textured images consisting of four and five textures

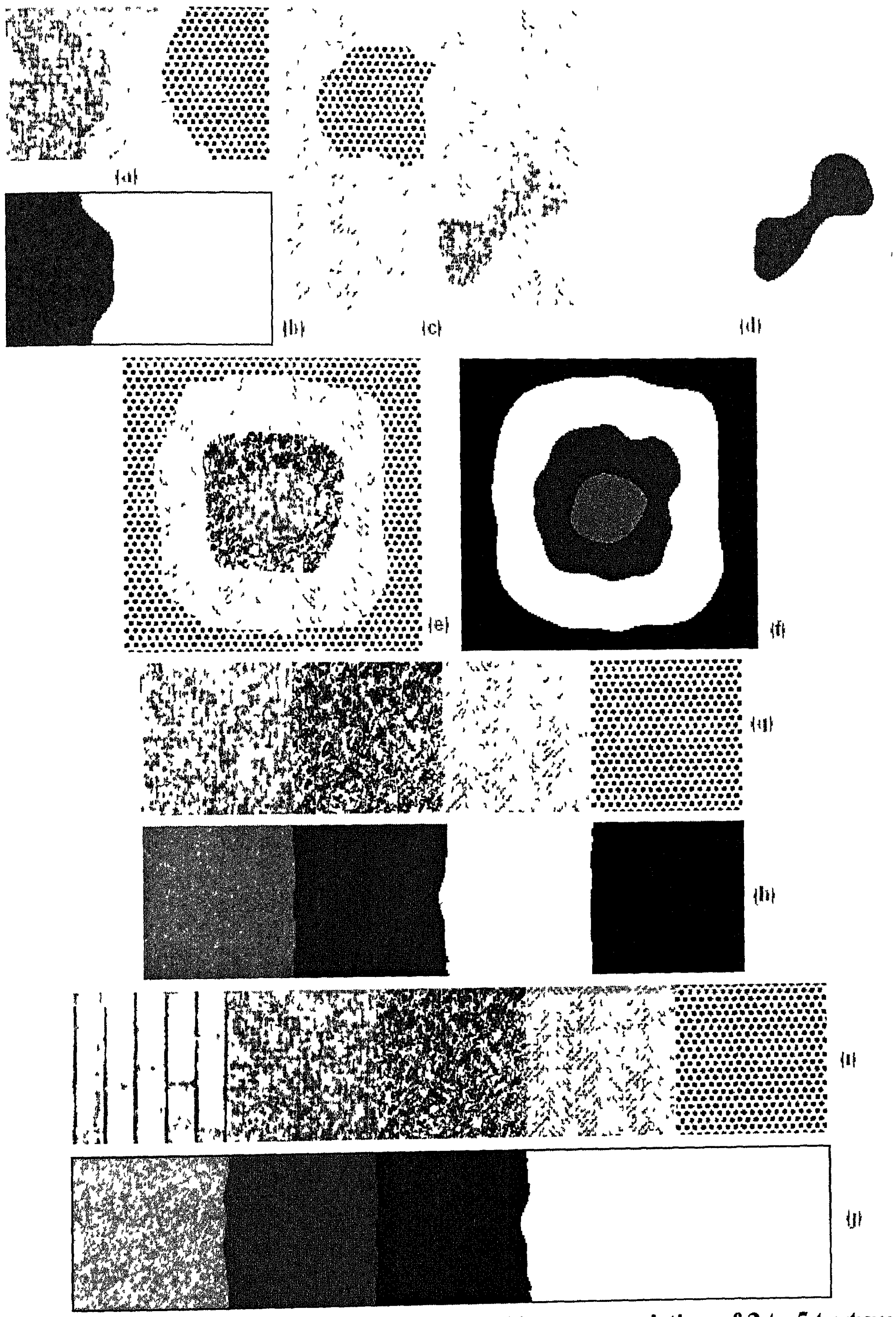


Figure 3.10 : Segmentation of Multi-textured images consisting of 3 to 5 textures

3.5 Conclusions

In this work on texture segmentation, we have not dealt with orientation invariance of texture classification This work could be extended by increasing the feature space by adding the features corresponding to Gabor filters oriented in certain directions It is observed from the results, that the segmented regions are not always having the exact edges as in the original image Hence, the exact edge localization [25] of the segmented image will be a good enough add on to this work More so, the work could also be extended for colored textures, where the texture analysis could be performed at the gray scale, while color classification could be dealt with another neural network classifier working in parallel to the RBFNN classifier

This work provides Mathematical and experimental evidence suggesting that the application of Gabor Filters to Textured Images produce certain characteristic output signatures that are useful for segmenting the image It is clear that in a truly autonomous Texture Segmentation Architecture, (such as the Human Visual System) filters can not be customized to individual Textures In principle, a bank of Filters is required that spans the spatial frequency domain of the textures of interest Here, we have shown results for classifying and segmenting multi-textured images consisting of three to five textures As the Real Coded Genetic Algorithms provide efficient search in the parameter space, only a few Gabor Filters can be efficiently designed for efficient segmentation and classification of multi-textured images It may not be always necessary to have the filter bank size same as the number of textures For a given set of Textures it may also be possible to design smaller filter bank for efficient discrimination Finally, the use of RBFNN classifiers enable proper texture classification even from overlapping feature clusters

Chapter 4

A Psychovisually Viable Similarity Measure

The primary task of any Content Based Image Retrieval System is associated with the estimation of the similarity between the query and the image database components In most of the cases, the pixelwise comparison is however not a solution to the problem The query processing should take care of the perceptual similarity or semantics associated with the objects of comparison The evaluation of a meaningful measure of similarity rests on two important factors – Representation of the object in a feature space that adequately encodes the characteristics that we intend to measure and defining a suitable similarity measure on the space

In a number of cases, after having selected the right set of features and having characterized the object as a point in a suitable vector space, Computer Vision researchers make an uncritical assumption about the metric of the feature space Generally, the distance functions are used as metrics in query processing More the distance between the query and the subject of comparison, higher is the dissimilarity between the two However, the Psychology literatures are not in agreement with the assessment of (dis) similarity judged by means of distance functions Psychologists have challenged [8], [26], [27], [9] the applicability of the Euclidean distance measure and have set out to alternate measures of similarity [8], [10]

Tversky [8] have suggested a similarity measure based on his famous Feature Contrast Model Santini et al [10] extend Tversky's measure on a Fuzzy Logic Framework However, there are a number of limitations in the implementations of the model This work proposes an alternative measure of similarity based on Fuzzy Inference System (Henceforth denoted as FIS) The FIS based measure is designed after a number of Psycho-visual tests and thus is viable enough from the point of psycho-visual perception

This chapter is organized as follows Section 4 1 reviews the similarity theories in relation to psycho-visual perception along with the existing measures of similarity In section 4 2, we discuss the theory of FIS in details Section 4 3 introduces our measure and the results are shown in section 4 4 Finally, section 4 5 draws the conclusions and suggests the future extensions of the work

4.1 Similarity Theories

In this section we present the theories of human similarity judgement introduced by the psychologists We also review the different similarity measures put forward by several researchers and discuss the merits and flaws of the various approaches Firstly, section 4 1 1 briefs the metric axioms related to similarity judgement and analyzes the theories set forth by psychologists for abandoning the same Section 4 1 2 explains the similarity axioms suggested by the psychologists in a perceptual perspective to describe the properties of a valid similarity measure Section 4 1 3 presents the Set-theoretic similarity measure introduced by Tversky [8] and the extension of the same as the Fuzzy Feature Contrast Model suggested by Santini et al [10] is discussed in section 4 1 4

4.1.1 The Metric Axioms

A number of similarity measures proposed in the Computer Vision literature explain similarity (or, more properly, dissimilarity) as a distance in some suitable feature space that is assumed to be a metric space A distinction is however made between perceived dissimilarity (D_P, say) and the judged dissimilarity (D_J, say) [26], which is accessible to experimentation If F_A and F_B are the representations of the objects A and B in the feature space, then $D_P(A,B)$ is the perceptual distance between the two, while the judged distance is given by,

$$D_J(A,B) = g[D_P(A,B)] \tag{4 1}$$

g being a suitable monotonically non-decreasing function

The objects (A, B etc) are represented as points in a metric space formed from a suitable number of their characteristic features and $D_P(A,B)$ is the distance function of that space The metric model postulates that the perceptual distance $D_P(A,B)$ satisfies the metric axioms, whose experimental validity is challenged by several researchers

The first property of any distance function in a metric space is,

$$D_P(A,A) = D_P(B,B) \tag{4 2}$$

which can be stated as the constancy of self-similarity Visual comparison of two objects depends on a number of factors like saliency, goodness of form etc Two different objects can have their own salient features and thus when compared with itself may not yield the same index of similarity The constancy of self-similarity has been refuted by Krumhansl [9]

A second axiom of the metric model is minimality, given by,

$$D_P(A,B) \geq D_P(A,A) \tag{4 3}$$

The second axiom states the minimality of self-similarity Tversky [8] argued that in some recognition experiments the off-diagonal entries of the confusion matrices exceed the diagonal ones thereby showing the violation of the second metric axiom

Thirdly, the metric axiom states the symmetry property of the perceived dissimilarity as,

$$D_P(A,B) = D_P(B,A) \tag{4 4}$$

Just as in the previous cases, this axiom is also subject to experimental verifications A number of investigators ([8], [26], [9]) have attacked this assumption with direct similarity experiments This phenomenon is often associated with the "saliency" or "goodness of form" associated with the objects under visual comparison Similarity judgements can be regarded as extensions of similarity statements For example, we say "The Son Resembles the Father" rather than "The Father Resembles the Son" In general the less salient object is more similar to the more salient (or more prototypical) one than the more salient object is similar to the less salient Tversky [8] have presented empirical evidences for asymmetric similarities and have argued that similarity should not be treated as a symmetric relation

The final metric axiom is the Triangle Inequality, expressed as,

$$D_P(A,B) + D_P(B,C) \geq D_P(A,C) \qquad (4\ 5)$$

The triangle inequality implies that if A is quite similar to B, and B is quite similar to C, then A and C cannot be very dissimilar from each other Thus, it sets a lower (upper) limit to the similarity (dissimilarity) in terms of the similarities (dissimilarities) between A, B and B, C Tversky ([8], [27]) gave examples for violation of the triangle inequality, which indicates that it should not be accepted as a cornerstone of similarity models

In conclusion, it appears that the geometric model of similarity analysis faces several difficulties The applicability of the metric assumptions is limited and questionable Specifically, self-constancy and minimality are somewhat problematic, symmetry is apparently false and the triangle inequality is hardly compelling In spite of all these difficulties, the metric model is still in use in the Computer Vision community However, the psychologists have abandoned the metric model of perceptual similarity and have developed a different approach to similarity based on

Feature Matching ([8], [27]) The next section reviews the alternative theoretical approach to similarity analysis and briefs the relations of the new model

4.1.2 The Monotone Proximity Structure

The distance axioms thus impose an unnecessary set of properties whose viability are questionable from the viewpoint of human judgement of similarity Tversky and Gati [27] identify three ordinal properties of the dissimilarity (similarity) measures on a more generalized framework and use them to replace the metric axioms in what they call the "Monotone Proximity Structure" Suppose, the stimulus A has two features x and y and is given by A≡xy Let, D(A,B) denote the dissimilarity between the stimuli and $D(x_1,x_2)$ denote the feature-wise dissimilarity measure A monotone proximity structure is characterized by three properties described as follows

Dominance

$$D(x_1y_1, x_2y_2) > \max\{ D(x_1y_1, x_1y_2), D(x_1y_1, x_2y_1) \} \qquad (4\ 6)$$

That is, the two-dimensional dissimilarity exceeds both the projections of the one-dimensional dissimilarities

Consistency

For all x_1, x_2, x_3, x_4 and y_1, y_2, y_3, y_4,

$$D(x_1y_1, x_2y_1) > D(x_3y_1, x_4y_1) \Leftrightarrow D(x_1y_2, x_2y_2) > D(x_3y_2, x_4y_2) \qquad (4\ 7)$$

$$D(x_1y_1, x_1y_2) > D(x_1y_3, x_1y_4) \Leftrightarrow D(x_2y_1, x_2y_2) > D(x_2y_3, x_2y_4) \qquad (4\ 8)$$

That is, the ordinal relation between dissimilarities along one dimension is independent of the other co-ordinates

Transitivity

We define x_2 to be between x_1 and x_3 if $D(x_1y, x_3y) > \max\{ D(x_1y, x_2y), D(x_2y, x_3y) \}$, and we write it as $x_1|x_2|x_3$ It is to be noted that in view of Consistency, "Betweenness" is well defined as it is independent of the co-ordinate y that appears in the definition

The third property of the monotone proximity structure needs the concept of Betweenness, which states that,

$$[x1|x2|x3] \text{ AND } [x2|x3|x4] \Rightarrow [x1|x2|x4] \text{ AND } [x1|x3|x4] \qquad (4\ 9)$$

This framework is more general than the metric model While all distance functions satisfy dominance, consistency and transitivity, not all the proximity structures satisfy the distance axioms Dominance is a week form of the triangle inequality that applies along the co-ordinate axes Consistency ensures that certain ordinal properties related to the ordering of the features x do not change when y is changed. Transitivity ensures that the "betweenness" relation behaves as in the metric model, at least when moving along the axes of the feature space Most of the distance measures proposed in the literature, as well as the feature contrast model (described in the next section), predict that dominance, consistency and transitivity hold, thereby establishing the validity of the proximity structure over the metric model

4.1.3 Set-Theoretic Similarity

In his 1977 paper [8], Tversky proposed his famous Feature Contrast Model Instead of representing objects as points in a metric space, Tversky characterized them as sets of binary features The development of the Set-Theoretic Similarity Measure is explained in details in Tversky's original paper [8] In this section we outline only the main result from his paper

Let, the object 'a' be characterized by the set A of features that it possesses Equivalently, a feature set is the set of logic predicates, which are true for the object in question Let a and b be two objects whose features are given by the sets A and B, respectively The similarity of the object a compared to b is defined by S(a,b) The main result of Tversky's paper defines this similarity function as -

$$S(a,b) = f(A \cap B) - \alpha f(A-B) - \beta f(B-A) \qquad (4\ 10)$$

Where, f is a non-negative set function and α, β are two non-negative constants $A \cap B$ denotes the features common to both the objects, whereas, A-B and B-A denote the distinctive features of a and b The model describes the similarity to be a function of the common features of two objects which gets reduced due to the distinctive features thereby justifying its name as the Feature Contrast Model This model also accounts for the asymmetric nature of similarity It is obvious from equation 10, that for different values of α and β, $S(a,b) \neq S(b,a)$ Setting $\alpha > \beta$, we introduce the directionality of similarity as $S(a,b) > S(b,a)$ [whenever, $f(A) < f(B)$] This implies that the direction of the asymmetry is determined by the relative "salience" of the object If 'b' is more salient than 'a', then 'a' is more similar to 'b' than vice versa

However, the serious problem in adoption of the feature contrast model lies in the implementation of the same In Tversky's theory, each object is characterized by the presence or absence of features This convention forces Tversky to adopt complex mechanisms for the representation of numerical quantities As for example, length, which is a positive quantity is discretized into a sequence L_i and represented as a collection of feature sets such that if $L_1 < L_2 < \quad < L_n$, then $A_1 \subset A_2 \subset \quad \subset A_n$ Quantities that can be either positive or negative are represented by even more complex constructions Thus the assumption of binary features would leave us with the problem of evaluating logic predicates based on some continuous and noisy measurements, yielding brittle and unreliable features More so, Tversky's feature contrast model applies to a particular type of features – those that can be expressed as predicates over the object domain Santini et al [10] suggests a solution to this problem by extending Tversky's work in a Fuzzy Set-Theoretic framework, where the use of Fuzzy Logic will allow the extension of Tversky's results to situations where modeling by enumeration of features is impossible or problematic The next section discusses the Fuzzy Feature Contrast Model in brief

4.1.4 The Fuzzy Feature Contrast Model

Consider the problem of judgement of the height of a person Typically, we comment on the height of a person as "The person is tall" or "The person is short-heighted" However, the comment about the person's height can be made with reference to some fixed height (say 5 5 feet), above which we call him a tall person This form of crisp judgement either assigns him as a tall or a short heighted person On the other hand, the new science of Fuzzy Logic does not agree with this form of reasoning and rather prefers implementing the same in qualitative terms like "Very Tall", "Very Short", "Little Taller" etc In the following sections we review the basic concepts related to Fuzzy Set Theory and discuss its application in extending Tversky's measure

4.1.4.1 Fundamentals of Fuzzy Logic

The classical Set Theory [28] defines a set to be a collection of distinct objects It is defined in such a way to dichotomize a given Universe of Discourse (Containing all elements, denoted by U) into two groups – members or non-members of a certain crisp set A classical crisp set (A) is also defined by the so called characteristic function The characteristics function $\mu_A(x)$ of a crisp set A in U takes its values in {0,1} and is defined as,

$$\begin{aligned} \mu_A(x) &= 1, x \in A \\ &= 0, x \notin A \end{aligned} \tag{4 11}$$

It is to be noted here that the crisp set A has rigid boundaries and sharply distinguish its members from its non-members Though, the Classical Set theory has become the foundation of Modern Mathematics, its application in implementing qualitative reasoning is however, limited by the fact that the human reasoning does not follow any such rigid decision boundaries

A fuzzy set, on the other hand, introduces vagueness by eliminating the sharp boundary that divides members from nonmembers in the group Thus, the transition between full membership and non-membership is *gradual* rather than abrupt A Fuzzy set B in the universe of discourse U is defined as a set of ordered pairs, given by,

$$B = \{ (x, \mu_B(x)) \mid x \in U \} \qquad (4\ 12)$$

Where, $\mu_B(\,)$ is called the *Membership Function* of the fuzzy set B and $\mu_B(x)$ denotes the degree of belongingness (membership) of x in B As for an example, let us consider the fuzzy set of "Tall Persons" A 6-ft tall person belongs to this set as well as person of height 4-ft However, the values of their membership really matter The *Taller* person belongs to a *higher degree* to the set as compared to the *short heighted* fellow Figure 4 1 shows the graphical representation of the crisp and fuzzy judgements for the sets "Short" and "Tall"

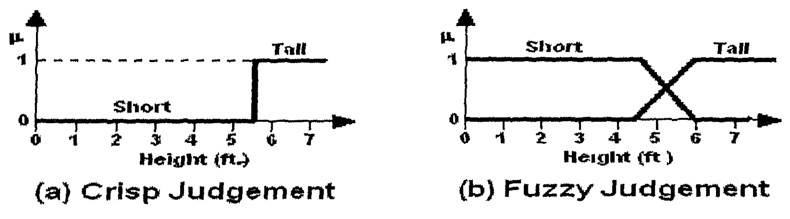


Fig. 4.1 : Graphical View of (a) Crisp and (b) Fuzzy Judgement

The membership function $\mu_B(\,)$ maps the universe of discourse U to the membership space M, i e $\mu \;\; U \rightarrow M$ When $M = \{0,1\}$, the set is non-fuzzy and μ is the characteristic function of the crisp set However, for fuzzy sets, the *range* of the membership function is a subset of the nonnegative real numbers whose supremum is finite In most general cases, M is set to the unit interval [0,1] Hence, fuzzy sets may be viewed as an extension and generalization of the basic concepts of crisp sets, however, some theories are unique to the fuzzy framework

The basic operations of crisp sets are also extended to define the same for fuzzy sets Let, P and Q be two fuzzy sets in the universe of discourse U The following define these basic operations on P and Q in the fuzzy set theoretic framework as,

1. **Complement:** We denote the complement of the set P as P' and the membership function is given by,

$$\mu_{P'}(x) = 1 - \mu_P(x), \ \forall x \in U \tag{4 13}$$

2. **Intersection:** The intersection of two sets P and Q is symbolically represented as P∩Q and the membership function of the resultant set is given by,

$$\mu_{P \cap Q}(x) = \min\{\mu_P(x), \mu_Q(x)\}, \ \forall x \in U \tag{4 14}$$

3. **Union:** The union of two sets P and Q is represented as PQ and the membership function of the resultant set is given as,

$$\mu_{P \cup Q}(x) = \max\{\mu_P(x), \mu_Q(x)\}, \ \forall x \in U \tag{4 15}$$

4. **Equality:** The two sets P and Q are said to be equal iff,

$$\mu_P(x) = \mu_Q(x), \ \forall x \in U \tag{4 16}$$

5. **Subsethood:** The set P is said to be the subset of set Q, represented as P⊆Q iff,

$$\mu_P(x) \leq \mu_Q(x), \ \forall x \in U \tag{4 17}$$

With this short introduction to Fuzzy Logic and Set Theory, we briefly discuss the extension of Tversky's Similarity measure in the next section

4.1.4.2 Redefining Tversky's Measure

Let Ω be a set and ϕ $\Omega \rightarrow R^m$ be a set of measurements on the elements of Ω Let $P(\omega)$ be a predicate about the element $\omega \in \Omega$ The truth of the predicate $P(\omega)$ is given by,

$$T\{P(\omega)\} = \mu\{\phi(\omega)\}, \text{ where, } \mu \quad R^m \rightarrow [0,1] \tag{4 18}$$

From the measurements ϕ, the truth values of p fuzzy predicates are derived and collected into a vector

$$\mu(\phi) = \{ \mu_1(\phi), \mu_2(\phi), \quad , \mu_p(\phi) \} \tag{4 19}$$

$\mu(\phi)$ is called as the fuzzy set of true predicates on the measurements ϕ The set is fuzzy in that a predicate P_j belongs to $\mu(\phi)$ to the extent $\mu_j(\phi)$

In order to apply the feature contrast model to the fuzzy sets ϕ and ψ, the non-negative function f is to be chosen properly along with defining means for computing the fuzzy sets $\mu(\phi)\cap\mu(\psi)$, $\mu(\phi)$-$\mu(\psi)$ and $\mu(\psi)$-$\mu(\phi)$ The non-negative function on the fuzzy set $\mu = \{\mu_1,\mu_2, \quad ,\mu_p\}$ is assumed to be given by the cardinality of the same, given by,

$$f(\mu)=\sum_{j=1}^{p}\mu_j \tag{4 20}$$

The set theoretic operations of intersection and difference on the fuzzy sets $\mu(\phi)$ and $\mu(\psi)$ are defined as,

$$\mu_{\cap}(\phi,\psi) = [\min\{\mu_1(\phi), \mu_1(\psi)\},\min\{\mu_2(\phi), \mu_2(\psi)\}, \quad , \min\{\mu_p(\phi), \mu_p(\psi)\}] \tag{4 21}$$

$$\mu_{-}(\phi,\psi) = [\max\{\mu_1(\phi)-\mu_1(\psi), 0\}, \quad , \max\{\mu_p(\phi)- \mu_p(\psi), 0\}] \tag{4 22}$$

With these definitions, Tversky's similarity function between two fuzzy sets $\mu(\phi)$ and $\mu(\psi)$ corresponding to measurements made on two images could be written as,

$$S(\phi,\psi)=\sum_{j=1}^{p}\min\{\mu_j(\phi),\mu_j(\psi)\}-\alpha\sum_{j=1}^{p}\max\{\mu_j(\phi)-\mu_j(\psi),0\}-\beta\sum_{j=1}^{p}\max\{\mu_j(\phi)-\mu_j(\psi),0\} \quad (4.23)$$

Though, the fuzzy feature contrast model successfully overcomes the feature representation problems of Tversky's model, still there are limitations from the implementation point of view. No indications are however given for the proper selection of the values of α and β, the prime parameters of the similarity measure. Similarity being a context sensitive issue, we should have proper choices of these parameters for definite Computer Vision related tasks. More so, a mere Mathematical model like those described above do not necessarily translate the visual perception into definite numerical values for efficient query processing. Thus, we set out to find an alternative measure for efficient quantitative description of the highly qualitative issue of similarity in a framework of human like reasoning. The proposed measure is described in details in section 4.3 after a brief review of the Fuzzy Inference System in the next section.

4.2 The Fuzzy Inference System

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic, thereby implementing the quantification of the qualitative human reasoning. The mapping then provides a basis from which decisions can be made regarding the system output. The performance of the FIS depends on the *Fuzzy IF-THEN Rulebases*, *Fuzzification of Inputs* and the *Output Defuzzification Procedure*. In section 4.1.4.1, we have discussed the basic concepts associated with

Fuzzy Logic and Fuzzy Set Theory In the following sections we explain the principle of operation of a typical FIS along with its main components

4.2.1 The Linguistic Variables

An ideal medium to characterize fuzziness is natural languages A *linguistic Variable* takes words or sentences from natural language expressions as its values Each linguistic variable has an associated *term set*, which denotes the set of linguistic values or labels that the variable can assume Each linguistic value corresponds to a fuzzy subset in appropriate domain As for example, if weight is interpreted as a linguistic variable, then its term set T(weight) is given by,

T(weight) = { "Very Fat", "Fat", "Medium", "Slim", "Very Slim", } (4 24)

Where, each term in T(weight) is characterized by a fuzzy set in a universe of discourse of, say, U = [0 150] Kg We might interpret "slim" as a weight below 70 Kg and "Fat" as above 90 Kg Thus, the membership functions of these fuzzy sets on U could be designed accordingly Typical values of linguistic variables contain *primary terms* such as "slim" or "fat" as well as *hedges* or *quantifiers* as "very", "Medium" or "Low" Thus, to design a FIS, firstly one should fix up the term set containing the "primaries" and "hedges" associated with the linguistic variables of the system and then to decide the input-output mapping based on fuzzy conditional statements

4.2.2 The Fuzzy IF-THEN Rules

In classical logic, the truth-value of the expression "IF A THEN B" (A→B) where A and B are propositional variables is defined by a *Truth Table* However, a more general concept, which plays an important role is the *Fuzzy Conditional Statement* or *The Fuzzy IF-THEN Rules* They are expressions of the form "IF P THEN Q", where

P and Q are labels of fuzzy sets charactenzed by appropnate membership functions These statements descnbe a relation between two different fuzzy subsets P and Q of disparate universes of discourse U and V respectively The IF part of the rule (IF P) is called the *antecedent* or *premise* and the THEN part (THEN Q) is said to be the *consequent* or *conclusion*

Due to their concise form, they are often employed to capture the imprecise mode of reasoning which plays an essential role in the human ability to make decisions in an environment of uncertainty and imprecision The following are typical examples of such statements

IF *pressure* is *very high* THEN *volume* is *low*

IF *temperature* is *medium* THEN *fan-motor speed* is *high*

Where, "pressure", "volume", "temperature" and "fan-motor speed" are linguistic vanables and the terms "very high", "low", "medium" and "high" are *linguistic values* or *labels characterized by their membership functions*

Generally, a fuzzy system works with more than one IF-THEN rules The collection of all the rules of the FIS is called the *Rulebase* More so, the rules may not be as simple as "IF A THEN B", rather they will take the general form as "IF X is A AND Y is B OR U is C THEN Z is E AND V is F" As for Example, let us consider a typical two input (X and Y, say) one output (Z, say) system defined over the universes of discourse U, V and W respectively Let the term sets associated with the fuzzy vanables be,

$$T(X) = \{ x_1, x_2, x_3 \}$$

$$T(Y) = \{ y_1, y_2 \}$$

$$T(Z) = \{ z_1, z_2, z_3 \} \tag{4 25}$$

And, we wnte the rulebase as,

Y \ X	x_1	x_2	x_3
y_1	z_1	z_2	z_3
y_2		z_2	z_3

Where, the rules (e g First row-Second column of the rule matnx) are read as "IF X is x_2 AND Y is y_1 THEN Z is z_2" Similarly, rule matrices could also be written for OR and other such relations It is also to be noted that for don't care conditions or non-existent rules, we may leave the space blank The rulebase *R* defines the mapping of the input space to the output space as $R \quad U \times V \rightarrow W$ The rules should be properly designed as it forms the backbone of the fuzzy system The fuzzy inference algorithm parallely activates all the rules of the rulebase for reaching a finalized decision

4.2.3 Fuzzification

Fuzzification means adding uncertainty by design to crisp sets It is a process of spreading the information provided by a crisp number or symbol to its vicinity, so that the closed neighborhood of the crisp number can be recognized by the computational tools This is however analogous to the interpolation method between two data points Fuzzification corresponds to defining interpolation surfaces between two sets or two numbers by a controlled (designed) uncertainty distribution function Practically, the injection of *designed uncertainty* enables us to carry on computations in those gray areas not explicitly defined by crisp entities

As for an example, consider the linguistic variable of "Length" on the universe of discourse U = [0, 10] cm having the Term set T associated with it as,

T(Length) = { "Very Short", "Short", "Medium", "Tall", "Very Long" } (4 26)

Figure 4 2 shows the membership functions associated with the members of the term set T The process of Fuzzification computes the membership values of the input variable in all the term set members Consider the fuzzification of a length value of 6 6 cm as shown in the figure It is seen that this value have memberships of 0 36 and 0 79 respectively in the membership functions of "Long" and "Medium" respectively and zero membership value in others Thus, fuzzifying the crisp value 6 6 cm we get the vector of membership function values as,

$$\mu_{Length}(6\ 6\ \text{cm}) = \{\ 0, 0, 0\ 79, 0\ 36, 0\ \} \qquad (4\ 27)$$

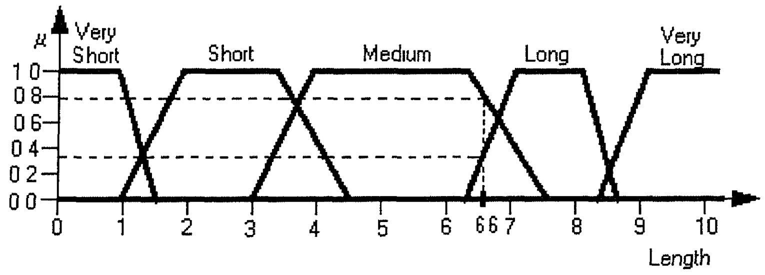


Fig. 4.2 : Fuzzification of Crisp Input

The fuzzy inference algorithm does not work directly on crisp input values, rather it operates on the fuzzified form of the same Thus, the concept of fuzzification is very important in context of the operating principle of the FIS

4.2.4 The Fuzzy Implication

Previously, we have discussed the basic concepts related to fuzzy rulebases and input fuzzification This section explains the fuzzy implication procedure for evaluating the truth-value of the IF-THEN rules in case of multiple input systems by combining the membership values of the antecedent fuzzy variables The composition of the antecedent variables may be achieved in a number of ways [29] Consider composing the rule of the form "IF X is x AND Y is y THEN Z is z" Let, the membership value for the antecedent statement “X is x” be m and that of “Y is y” be n Let, p be the truth-value of the statement “Z is z” evaluated through fuzzy implication

The minimum value method proposed by Mamdani and the product value composition procedure set forth by Larsen are the widely used ones for AND type of composition [29] They are defined by the following equations

Minimum value method: $p = \min\{ m, n \}$ (4 28)

Product value procedure: $p = m \times n$ (4 29)

For the OR type of composition, the maximum value composition (Mamdani) or the algebraic sum method (Larsen) are used, which are defined as,

Maximum value composition: $p = \max\{ m, n \}$ (4 30)

Algebraic Sum Method: $p = m + n$ (4 31)

Here, we have stated the compositional procedures for two input systems only However, these could be extended similarly for dealing with rules having more than two statements and a composite AND-OR nature

4.2.5 Fuzzy Rule Aggregation

In the previous section, we have discussed the procedure of fuzzy implication for a single rule However, there are a number of rules in a FIS rulebase that may be having the same consequent Thus, further we need to combine the truth-values of the similar consequents to reach at a final conclusion This process of combining the individual implication results is called the fuzzy rule aggregation

In order to explain this procedure by an example, we refer to the two input one output system described in section 4 2 2 Let us consider the case of presenting the crisp values u and v ($u \in U$ and $v \in V$) to the fuzzy system under consideration, which after the procedure of input fuzzification (described in section 4 2 3) yields the fuzzified membership value vectors as,

$\mu_X(u) = \{ m_1, m_2, m_3 \}$, and

$\mu_Y(v) = \{ n_1, n_2 \}$ (4 32)

Let the element of the ı'th row and ȷ'th column of the rulebase *R* denote the rule R_{ij} More so, we assume the fuzzy ımplıcatıon procedure for AND to be the mınımum value method Then we can tabulate the truth-values of the consequents of the dıfferent rules as,

Table 4.1: Illustration of Fuzzy Rule Aggregatıon

Rules / Consequent	R_{11}	R_{12}	R_{13}	R_{22}	R_{23}
"Z ıs z_1"	$\min(m_1,n_1)$				
"Z ıs z_2"		$\min(m_2,n_1)$		$\min(m_2,n_2)$	
"Z ıs z_3"			$\min(m_3,n_1)$		$\min(m_3,n_2)$

Now, we reach the fınal fuzzy ınference vector by sımply addıng up the truth-values of the sımılar consequents However, thıs procedure of sımple addıtıon may exceed unıty, ı e the result may go out of the range In that case, we sımply replace the result by unıty Thus, ıf we denote the result of the fuzzy rule aggregatıon for the fuzzy varıable Z ın vector form as $\mu(Z) = \{p_1, p_2, p_3\}$, then for the system under consıderatıon we can wrıte,

$p_1 = \min(m_1,n_1)$,

$p_2 = \min\{\min(m_2,n_1) + \min(m_2,n_2), 1\}$, and

$p_3 = \min\{\min(m_3,n_1) + \min(m_3,n_2), 1\}$ (4 33)

4.2.6 Defuzzification

The fınal stage of the fuzzy ınference algorıthm ıs the process of defuzzıfıcatıon The procedure of fuzzy ımplıcatıon gıves us the fuzzy ınference vectors formed of the actıvatıons of the output varıable values However, to apply ıt meanıngfully ın real world applıcatıons, we need to convert the fuzzy result of the FIS ın crısp form Thıs process of removing fuzziness through conversion of fuzzy values to crisp values is termed as defuzzification

There are a number of defuzzification methods [29] proposed by the fuzzy system researchers These typically include the different centroid methods like *centre of gravity*, *centre of largest area*, *centre of weights* and the maxima methods like *mean of maximums*, *maximum possibility* etc In this section we will concentrate on the defuzzification methods based on the centroid computation

For the purpose of illustration, we consider the fuzzy system referred in sections 4 2 2 and 4 2 4 In the previous section we have computed the fuzzy inference vector $\mu(Z) = \{p_1, p_2, p_3\}$ for the crisp inputs u and v Fig 4 3 (a) shows the membership function distributions (for z_1, z_2 and z_3) of Z over its universe of discourse W The elements of $\mu(Z)$ denote the net activations of each of the membership functions of Z To represent the contribution of activation of each of these membership functions to the final result, we cut off the distributions at their respective activation values The procedure of cutting off the distributions and obtaining the partially activated membership functions is shown in figure 4 3 (b). Finally, we combine all the activated membership function by taking their union and obtain the shaded area as shown in figure 4 3 (c)

The centre of gravity method proposes the abscissa of the centroid of the shaded area to be the required crisp output value (w_c, say) The centre of gravity (w_c) of the area A is computed by the following equation

$$w_c = \frac{\int_A w\,dA}{\int_A dA} \qquad (4\ 34\ )$$

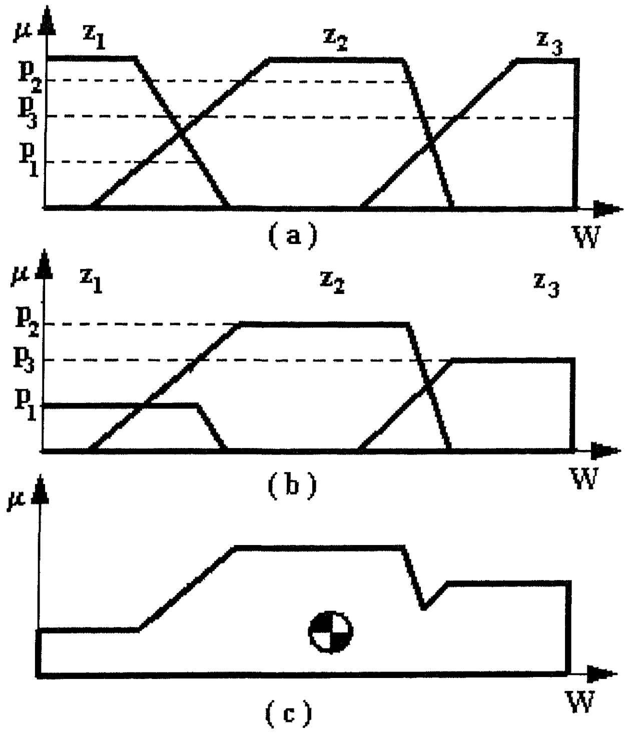


Fig. 4.3 : The steps of defuzzification using centre of gravity method

A second popular method for defuzzification is the centre of weights method, where, we assign some weights to the results of the implications of the individual rules This is something like taking advises on a problem from a number of experts and combining their conclusions by assigning weights on each advise according to the reliability of the advisor Let there be N rules of the system (R_1, , R_i, R_N) with the respective weights as ω_1, , ω_i, , ω_N Suppose, the implication result of the i'th rule R_i is p_i Let A_i be the activated area for the implication result whose centre of gravity is C_i Here, we don't go for the rule aggregation method described in section 4 2 5 Rather, we directly evaluate the output crisp value w as,

$$w = \frac{\sum_{i=1}^{N} \varpi_i C_i A_i}{\sum_{i=1}^{N} \varpi_i A_i} \qquad (4\ 35)$$

Between the two defuzzification methods described in this section, the centre of gravity method is more intuitively correct rather than the centre of weights method. However, the later introduces the concept of rule weights, which is also very important. Thus, we go for combining the advantages of both the methods and propose a new method of centroidal defuzzification with weighted fuzzy implication in a unified framework. We consider the example of the same rulebase described for the centre of weights method. Suppose the rule R_i has the consequent as "Z is z_k" and the fuzzy implication result for the rule is p_i. Let A_i be the area, when the membership function corresponding to z_k is cut-off at p_i. From the centre of gravity method, we infer that the cut-off area denotes the activation of an output fuzzy predicate, which represents the contribution of the implication result. Now, if the rule weight is ω_i, then the corresponding weighted rule activation should yield the area A'_i, given by,

$$A'_i = \omega_i A_i \tag{4.36}$$

Now we search for that value p'_i for which, when cut-off, the membership function corresponding to z_k will yield the area A'_i. This value p'_i is now taken as the implication result of the rule R_i and we proceed by rule aggregation procedure (as described in section 4.2.5) to arrive at the defuzzified output crisp value w_c through the centre of gravity computation (as per equation 4.34). Figure 4.4 (a), (b) and (c) describes the entire defuzzification procedure discussed above.

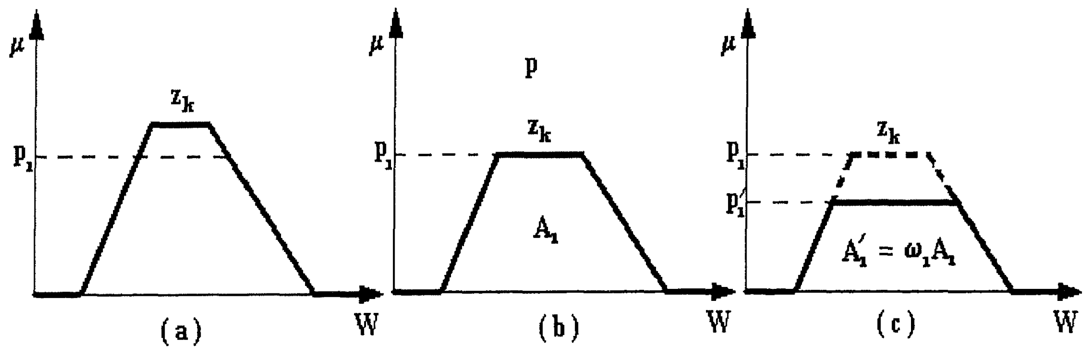


Fig. 4.4 : The new method for weighted fuzzy implication

4.2.7 Summary

In the previous sections, from 4 2 1 to 4 2 6 we have discussed the main components and the operating principles of a typical FIS In this section we summarize the operation of the FIS The crisp values are given as inputs to the FIS, which after fuzzification goes through the rulebase, where from the final fuzzy inference is drawn through the procedures of fuzzy implication and rule aggregation Finally, we compute the output crisp value through a suitable method of defuzzification Figure 4 5 shows a schematic diagram explaining the operational flow of the fuzzy inference algorithm

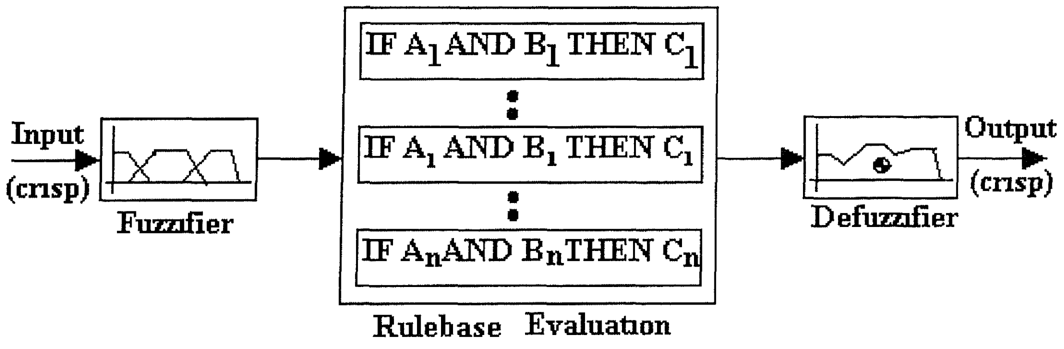


Fig. 4.5 : Schematic Diagram of Fuzzy Inference Algorithm

4.3 The FIS based Similarity Measure

The visual comparison of two objects is inherently associated with the concepts of similarity or dissimilarity between the two The human mind holds an abstract idea of the perceptual features of the objects under comparison and the results are expressed in terms of natural language statements like "Very Low Similarity", "Low Dissimilarity", "More or Less Similar" etc The main aim of an Image Retrieval System is to properly apply these qualitative notions of similarity (dissimilarity) while executing the query processing In section 4 1, we have reviewed the similarity theories from the psychology literatures, where we have seen that most of the works done so far are mainly concerned with the mathematical modeling of the perception of similarity and doesn't take into account the qualitative nature of visual perception

Hence, a proper similarity measure should be able to translate the qualitative reasoning of visual comparison to quantitative terms In section 3, we have discussed the theory and operation of Fuzzy Inference System, which arrives at quantitative conclusions through a set of associated linguistic statements Thus, we propose a novel method for computing the index of similarity (dissimilarity) by using fuzzy inference systems

Consider the case of query processing for a set of objects (O), which could be adequately represented as points in the same N-dimensional feature space F Let the query object (O_q) and that under visual comparison (O_c) be represented by the respective feature vectors as, $F_q = \{ f_{q1}, \ \ , f_{qi}, \ \ , f_{qN} \}$ and $F_c = \{ f_{c1}, \ \ , f_{ci}, \ \ , f_{cN} \}$ Now, we design 2N number of FIS for featurewise comparison, i e we have two fuzzy inference systems for each feature – one for judging similarity index and the other for dissimilarity Many researchers ([8], [10]) directly compute similarity (disimilarity) index simply by negation or some linear operation from the dissimilarity (similarity) one However, there are cases where human judgement looks for dissimilarity rather than similarity or vice versa while processing some query This clearly indicates the non-linear nature between the indices of similarity and dissimilarity and hence the evaluation of both of them should be done separately The design of the components of each FIS (viz Membership function distribution of variables over the universe of discourse and the weighted rulebase) is very much specific to that feature only The FIS are designed on the basis of a number of psycho-visual experiments, so as to properly encode the similarity (dissimilarity) judgements provided by human visual perception

Let, the similarity FIS be denoted as S_{FIS} and that of dissimilarity as D_{FIS} The value of the linguistic variable for the feature under consideration (e g "length", "colour" or "angle") is normalized to the universe of discourse [0,1] The similarity or dissimilarity indices are also defined on the universe of discourse [0,1] For both the FIS we apply the *centre of gravity* method of defuzzification, where the abscissa of the centroid gives the similarity (or dissimilarity) judgement Hence, we can write the results of the comparison of the i'th features as,

$$S_i = S_{FIS}(f_{qi}, f_{ci}) \tag{4 37}$$

$$D_i = D_{FIS}(f_{qi}, f_{ci}) \tag{4 38}$$

Where, S_i and D_i are the respective indices of similarity and dissimilarity The computation of the final measure of similarity (dissimilarity) is however expressed as a weighted average of the feature-wise similarity (dissimilarity) indices These weights are assigned from the results of the psycho-visual tests to indicate the saliency of distinctive features Thus, we can write the final measures for similarity and dissimilarity as,

$$S(O_q,O_c)=\frac{\sum_{i=1}^{N} S_i W_i}{\sum_{i=1}^{N} W_i} \tag{4 39}$$

$$D(O_q,O_c)=\frac{\sum_{i=1}^{N} D_i W_i}{\sum_{i=1}^{N} W_i} \tag{4 40}$$

Where, $S(O_q, O_c)$ and $D(O_q, O_c)$ denote the final similarity and dissimilarity indices for the comparison of the objects and the visual comparison result is expressed by the two-tupple (S,D) It ıs to be noted here that whıle wrıtıng the sımılarıty measure $S(O_q, O_c)$, as a conventıon we wrıte the query fırst and then the object under comparıson wıth the query object

4.4 Implementing the Similarity Measure

The performance of the newly proposed sımılarıty measure ıs valıdated on a set of sımple two-dımensıonal geometrıc objects where the judgement ıs made based on the shape features of the objects The prımary shape features assumed here are the length and angular ınclınatıon In thıs sectıon, we present the result ın two parts Fırstly, we descrıbe the procedure to desıgn the FIS components accordıng to human perceptıon of length and angular ınclınatıon and then we present the results of ımplementıng the measure along wıth the comparıson wıth actual sımılarıty judgement provıded by psycho-vısual tests

4.4.1 Designing the FIS

The membershıp dıstrıbutıons for length or angular ınclınatıon are decıded through the human perceptıon of the terms lıke "Very Low Length" or "Medıum Angular Inclınatıon" etc Typıcally for both the lınguıstıc varıables "Length" and "Angular Inclınatıon", we consıder the term set T as,

$$T(\bullet) = \{ \text{"Very Low"}, \text{"Low"}, \text{"Medium"}, \text{"High"}, \text{"Very High"} \} \qquad (4\ 41)$$

We present a lıne of length 100 mm and a quarter of a cırcle (90^o), on whıch fıve persons of dıfferent age group (who have no prevıous knowledge about our actıvıty) are asked to mark the segments for the members of the term set The subjects are also asked to mark the portıon about whıch they are confıdent to be "Very Low" or "Medıum" (marked by rectangular braces [ ]) along wıth the extent to whıch they can extend these notıons (marked by curved braces ( )) In fıgure 4 6, we show a typıcal result of the psycho-visual test performed for determining the membership function distribution

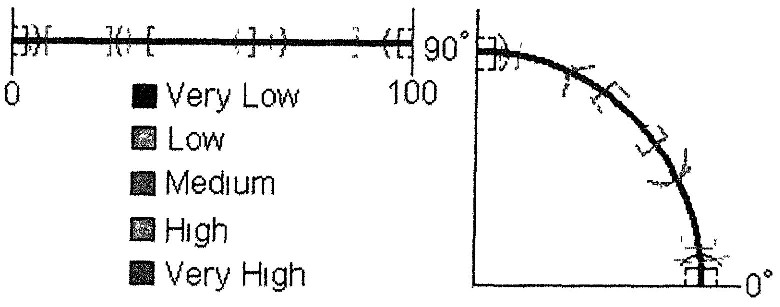


Fig. 4.6: Typical Result of Psycho-visual Test for Determination of Membership Function Distribution

We decide the finalized range of the members of the term set by taking the average of the five results Now, as the linguistic variables are defined over the universe of discourse [0,1], we normalize the ranges of the membership functions with respect to their maximum values, i e dividing the range parameters by 100 for "Length" and by 90° for "Angular Inclination" The membership function distribution for a term set member is typically assumed to be a flat headed one over the range of confidence for that member and with Gaussian tails on the extended parts The spread parameters of the Gaussian tails are determined by dividing the extension range by 3 Thus, if we have the confidence range as [a,b] with the respective left and right side extensions as [c,a] and [b,d], then the membership function is defined as,

$$\mu(x) = 1, \quad x \in [a,b]$$

$$= \exp[-(x-a)^2/\sigma_1^2], \quad x \in [c,a] \text{ and } \sigma_1 = (a-c)/3$$

$$= \exp[-(x-b)^2/\sigma_2^2], \quad x \in [b,d] \text{ and } \sigma_2 = (d-b)/3 \qquad (4\ 42)$$

We assume the same term sets and membership functions for the linguistic variables of "Similarity" and "Dissimilarity" over the universe of discourse [0,1] Since, there are no means of determining the membership function parameters for these variables through psychological tests, we simply assign the average of the range parameters for "Length" and "Angular Inclination" to that of "Similarity" and "Dissimilarity" In figure 4 7, we show the distributions of the membership functions for the variable "Length" over its universe of discourse

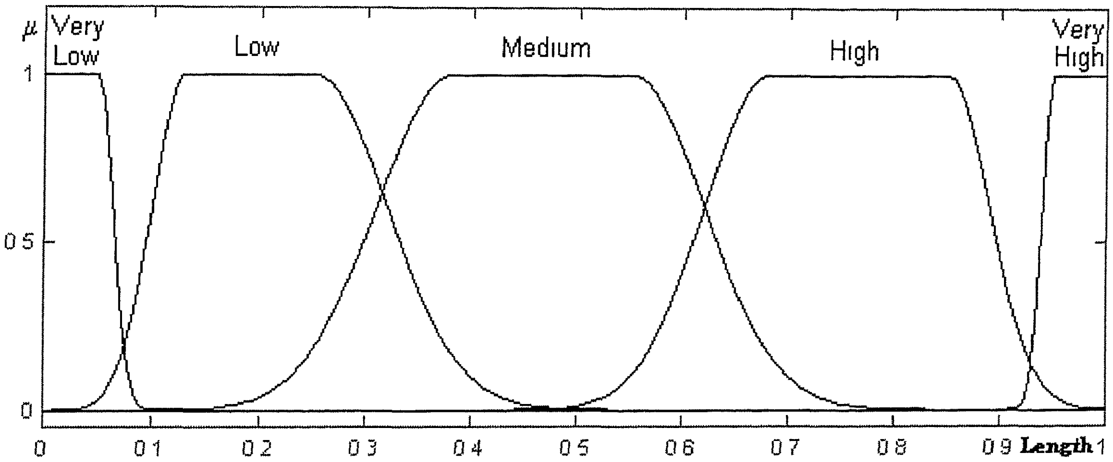


Fig. 4.7 : Membership Function Distribution For "Length"

Finally, we go for designing the FIS rulebase whose components will be of the form "IF Length(query) is Low AND Length(object under comparison) is Very Low THEN Similarity is High" and is weighted with a value between 0 and 1 For this purpose, we take one length (or angular inclination) value from the confidence range of each term set member as query and two values from the same as objects under comparison The query sets and the sets of comparison objects are presented to ten subjects The subjects are first asked to judge the similarity of the objects under comparison with respect to fixed queries and then the dissimilarity between the two We determine the rule bases of length-length and angle-angle comparison from the results of this psycho-visual test Suppose, the query is from the confidence range of "Medium" and the object under comparison is from that of "High", where 8 out of 10 judge the dissimilarity as "Low" Then we form the rule as "IF Length(query) is Medium AND Length(object under comparison) is High THEN Dissimilarity is Medium" and set the weight of the rule as 0 8 Appendix D lists the four rulebases for the line-line and angle-angle comparisons, having one similarity and one dissimilarity rulebase for each type of comparison We also plot the rule-surfaces, generated by the respective rulebases

4.4.2 Results of Implementation

In this section we present the results of implementing the measure and crosscheck the same with psycho-visual tests We validate our measure on the simple geometric figures like Rectangles, Parallelograms and ellipses Seven figures from each group are used to form the set of objects under comparison and three for queries Appendix E shows the figures used for the tests Five subjects were asked to rank the objects according to decreasing similarity with respect to fixed queries along with the features that they note for judgement of similarity

However, the results of ranking varied from person to person For the purpose of determining a final similarity ranking we take all the figures that have occurred in the similarity class and assign some weights to them The first ranking figure is given the highest weight say 7 and the next 6 and so on Each weight is multiplied with the number of occurrence of that figure in that rank The weighted sums for each figure are then arranged to get the final ranking As for example, we form the following table for analyzing the results

Table 4.2: Finalized Ranking for Psycho-visual Test Results

Q	1 (7)	2 (6)	3 (5)	4 (4)	5 (3)	6 (2)	7 (1)	Total
S1			5					25
S2				1	1	3		13
S3		5						30
S4					4		1	13
S5	5							35
S6				1		1	3	9
S7				3	2			15

Thus from the "total" column, we obtain the final ranking as S5, S3, S1, S7, S4, S2 and S6 However, it was found that both S2 and S4 had the same totals i e 13 In these cases we examine the rank distribution histograms of the figures and the one having its histogram more skewed towards higher ranks is given preference

In Appendix E we list the figures used as queries and objects of comparison In this section, we only list the ranking obtained from the psycho-visual tests and the results of applying the new similarity measure on the features suggested by the subjects and their priorities For comparison purposes, we also list the results for two commonly applied similarity measures, viz Euclidean Distance and Taxicab Norm

In case of rectangles, the width, height, area and the ratio of width to height (or, height to width, whichever is less than one) are taken as the primary features and all the features are assigned same weights Table 4 3 lists the results for rectangles

Table 4.3: Similarity Measures on Rectangular Figures

RANK	1	2	3	4	5	6	7	Ranking Error
				PSYCHO-VISUAL TEST				
RQ1	5	6	4	2	7	1	3	NA
RQ2	3	1	5	6	7	4	2	
RQ3	2	6	4	5	3	7	1	
				EUCLIDEAN DISTANCE				
RQ1	4	5	3	2	7	6	1	1 7143
RQ2	4	2	7	6	1	5	3	3 4286
RQ3	4	3	1	5	7	6	2	2 8571
				TAXICAB NORM				
RQ1	4	5	3	2	7	1	6	1 7143
RQ2	4	2	6	7	5	1	3	3 4286
RQ3	4	3	1	5	7	6	2	2 8571
				OUR MEASURE SIMILARITY FIS				
RQ1	3	1	6	5	2	7	4	2 8571
RQ2	3	1	5	6	2	7	4	0 5714
RQ3	2	6	7	5	3	1	4	1 1429
				OUR MEASURE DISSIMILARITY FIS				
RQ1	3	5	6	1	2	7	4	2 2857
RQ2	3	1	5	6	2	7	4	0 5714
RQ3	2	6	5	7	3	4	1	0 8571

For the ellipses, we use the features as the major and minor diameters, the enclosed area and the ratio of the diameters (whichever is less than one – major to minor or vice versa) Here, also we give same weightages to all the features Table 4 4 shows the comparisons of the results for ellipses

Table 4.4: Similarity Measures on Ellıptıcal Fıgures

RANK	1	2	3	4	5	6	7	Rankıng Error
PSYCHO-VISUAL TEST								
EQ1	5	3	1	7	4	2	6	NA
EQ2	6	2	4	7	1	3	5	
EQ3	4	7	2	6	1	3	5	
EUCLIDEAN DISTANCE								
EQ1	6	4	2	7	1	3	5	3 4286
EQ2	5	3	1	7	4	2	6	3 4286
EQ3	5	3	6	1	2	7	4	3 4286
TAXICAB NORM								
EQ1	6	4	2	7	1	3	5	3 4286
EQ2	5	3	1	7	4	2	6	3 4286
EQ3	5	3	6	1	2	7	4	3 4286
OUR MEASURE SIMILARITY FIS								
EQ1	5	3	1	7	2	4	6	0 2857
EQ2	6	4	2	7	1	3	5	0 2857
EQ3	4	7	6	2	1	3	5	0 2857
OUR MEASURE DISSIMILARITY FIS								
EQ1	5	3	1	7	2	4	6	2 2857
EQ2	6	4	2	7	1	3	5	0 5714
EQ3	4	7	6	2	1	3	5	0 8571

In case of parallelograms, the features are selected to be ıts base, heıght, base angle and area As suggested by the subȷects, here also we assıgn same weıghts to all the features In table 4 5 we lıst the results of applyıng the dıfferent measures

Table 4.5: Similarity Measures on Parallelogram Fıgures

RANK	1	2	3	4	5	6	7	Rankıng Error
PSYCHO-VISUAL TEST								
PQ1	4	7	2	6	5	3	1	NA
PQ2	6	7	4	5	2	3	1	
PQ3	3	2	4	6	7	5	1	
EUCLIDEAN DISTANCE								
PQ1	1	6	3	7	5	4	2	3 1429
PQ2	5	2	4	7	3	1	6	2 2857
PQ3	5	2	1	4	7	6	3	2 5714
TAXICAB NORM								
PQ1	1	6	3	7	5	4	2	3 1429
PQ2	5	2	4	7	3	1	6	2 2857
PQ3	5	2	1	4	7	6	3	2 5714
OUR MEASURE SIMILARITY FIS								
PQ1	5	4	7	6	2	1	3	1 4286
PQ2	6	4	7	1	3	5	2	1 4286
PQ3	3	6	2	1	4	7	5	1 4286
OUR MEASURE DISSIMILARITY FIS								
PQ1	5	4	6	7	2	1	3	1 7143
PQ2	6	4	7	1	3	2	5	1 4286
PQ3	3	6	1	2	4	7	5	1 7143

For the purpose of computation of the ranking error, we proceed in the following way Let us suppose that for the query q, the i'th rank (original rank obtained from psycho-visual tests) is predicted as the rank j(S,i) by the similarity measure (S), then the ranking error e_q (for Q number of comparison objects) is given by,

$$e_q = \frac{1}{Q}\sum_{i=1}^{Q}\left|i - j(S,i)\right| \tag{4 43}$$

From the above results, presented in tables 4 3, 4 4 and 4 5, it is seen that the widely used distance measures fail in all the cases, even in retrieving atleast the most similar figure On the other hand, the performance of our measure is quite appreciable in the sense that, even though it is designed based on judgements of a few number of persons, it has achieved the retrieval of atleast the first three similar figures (rectangles and ellipses) for both the similarity and dissimilarity FIS Though, the performance of the measure for ranking parallelograms is not that satisfactory, still it has superceded the distance measures through achievement of lower ranking errors

4.5 Conclusions

In this chapter, we have reviewed the similarity theories and compared the performance of different measures on a perceptual framework The newly proposed similarity measure based on human similarity perception is explained in details along with the implementation details It was found to perform better than the distance measures, which are widely used in the Computer Vision community The similarity measure was designed on the basis of the judgements of a very few number of subjects and thus, its performance will definitely improve, when tested on many

The weightages of features are assigned according to subjects' judgements However, the proper choice of feature weights would be a very important extension to this work The saliency is however, not expressed in any particular feature in all objects Rather, the factor of saliency comes into existence whenever very high similarity or contrast is observed in some feature(s) Typically, a weight function can be assumed over the range of similarity (dissimilarity) index i e [0,1] and could only be chosen if the test is performed on a large number of subjects, which was however one limitation in this work

Here, we have only considered the case for geometrical shapes However, there are other important components of visual perception like texture, colour, depth etc on which the measure could be extended In case of textures, sinusoidal gratings of various frequencies and orientations could be shown to the subjects, on the basis of which, we could design the FIS and the Gabor filters whose outputs will be taken as features, as they closely resemble the human receptive fields [11] Similarly, for colours we can also perform psycho-visual tests to design colour-similarity indices

So far this measure is referred in context to psycho-visual perception However, this measure is inherently powerful to encode any form of perceptual similarity as it is based on natural language judgements Thus, besides vision applications, it may be extended to judge similarity of stimuli perceived by other senses of human beings

Chapter 5

Conclusions and Future Extensions

In this thesis we have demonstrated an implementation of a prototype of Online Steel Surface Inspection System for Identification of visual surface defects on cold rolled steel sheets The inspection system is hosted on a single Intel Pentium Class Processor and the software modules are implemented in Visual C++ The prototype system however operates at a much lower speed of 1m/sec This section briefly discusses the conclusions, limitations and possible scopes for future extensions of the work

The proposed system concentrates on identification of four classes of surface defects, viz Black Patch, Anneal Colour, Indentation Marks and Holes along with perfect defect-free steel Among these, the human observers identify the Black Patch and Anneal Colour by their distinctive colour characteristics Following the human intuition, we go for extracting the features of colour information of these types of defects The results show that the defect region classification through colour feature extraction is quiet successful in identifying these defects The histogram features extraction also takes care of defect region rotations, as the histogram is rotation invariant More so, the application of Artificial Neural Networks in Classification tasks has accelerated the performance of the defect identification system However, the ANN approach of defect region classification through colour features is highly dependent on the imaging setup Thus, it depends on camera position and illumination, changing which needs to re-train the ANN classifier

Swellings or depressions on the steel surface characterize the indents The procedure of detection of indents thus tries to estimate the surface profile through the

Working at higher resolutions with increased computational power will enable us to implement computation intensive algorithms for texture feature extraction and analysis of shape or edge features Extending the feature set will be helpful in identification and modeling of new classes of defects The texture analysis and extraction of its features from Gabor filter outputs for the classification of the same have been dealt with in chapter 3 Textural features will be of much use in cases of defects like pinch mark, rust etc, which are rich in textural characteristic Thus, the inspection system performance could be enhanced for detection of a larger class of defects

In chapter 4, we have introduced a novel method for computing similarity indices based on judgements provided by human visual perception Till now the industry has been performing the inspection manually and thus hosts a number of experts in classifying steel surface defects Hence, a FIS based similarity metric could be designed by performing psycho-visual tests on these experts These tests could also suggest the distinctive features to be taken care of for representing some defect The fuzzy systems work faster than any other and don't take much time to learn – it's just a matter of adding some rules to the rulebase Thus, online learning capability could also be introduced in this system Thus, implementing the alternative similarity measure would be much faster and more inclined to the judgements provided by the defect inspection experts

Eventually, we have to design a shared memory parallel processing architecture interfaced with a number of cameras The use of multiple cameras enables the system to focus on the same region with higher resolution The increase in number of processors will add up to the computational power of the system The image data will be divided into the processors, each of which will be executing the same defect detection algorithms on their part of the data Thus, the use of parallel imaging hardware along with faster algorithms on extended defect feature space will accelerate the whole process of online inspection thereby enabling the system to perform in real time on a larger class of defects at higher speeds (30 frames per second for steel sheets moving at 30 m/sec) and resolution

Appendix A

This section describes the ***K-means clustering algorithm*** for finding the data clouds in a given data set Let us consider the problem of distributing M feature vectors into K clusters in a N dimensional feature space Let, X_{ij} denote the j'th component of the i'th vector, where, $1 \leq j \leq N$ and $1 \leq i \leq M$

Here, we label each vector to be belonging to a definite cluster, according to the minimum distance classification algorithm The distances (eg Euclidean distance) of each vector from the K cluster centres are computed, and the vector is assigned to the cluster having its centre at a minimum distance The K-means clustering algorithm is thus an iterative procedure of vector labeling, till a stable cluster assignment is achieved Figure A 1 shows a schematic flowchart of the algorithm

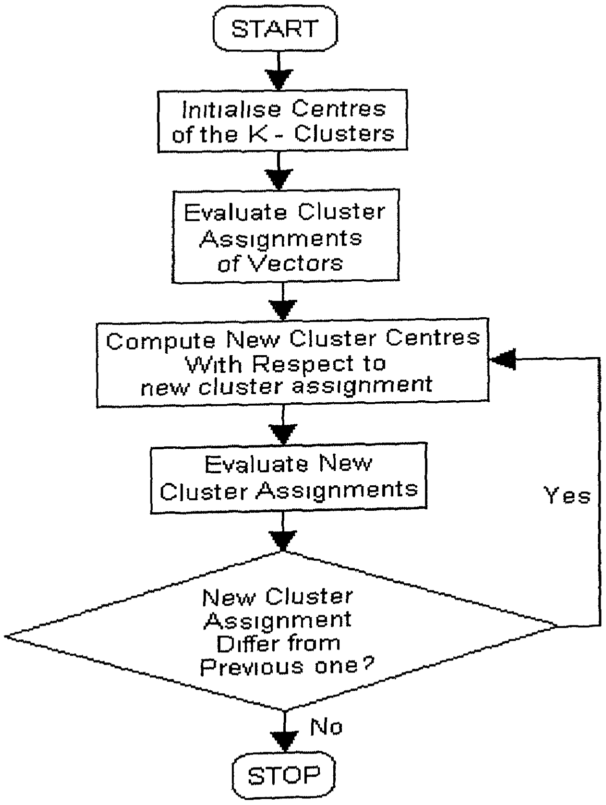

START
Initialise Centres of the K - Clusters
Evaluate Cluster Assignments of Vectors
Compute New Cluster Centres With Respect to new cluster assignment
Evaluate New Cluster Assignments
Yes
New Cluster Assignment Differ from Previous one?
No
STOP

Fig. A.1 : Flowchart of K-Means Clustering Algorithm

Generally, the components of the K initial centres are computed as K arithmetic means between the vector components To start with, we find a cluster assignment based on the initial cluster centers The cluster centres are updated from the newly obtained cluster assignment by computing the centroids of the cluster members The vectors are again labeled according to the updated cluster centres This procedure of cluster centre updating and vector labeling continues till the cluster assignment becomes a stable structure and does not change any more

Appendix B

This *Gradient Descent Algorithm* [30] is one of the most widely used approaches to solve optimization problems The algorithm is so called due to its dependency on search-surface gradient and the descending nature of the travel of the solution point co-ordinates Figure B 1 explains the principle of operation of the Gradient Descent Algorithm

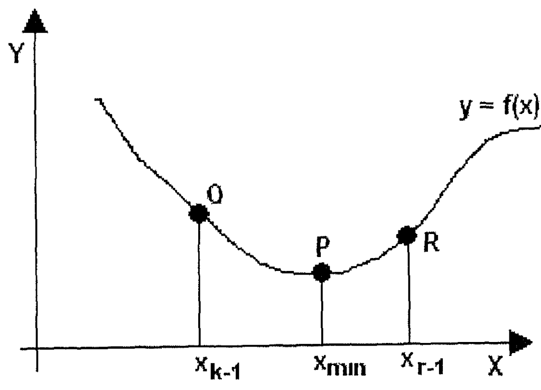


Fig. B.1: The Graphical Explanation of the Gradient Descent Algorithm

Consider the problem of reaching the minimum value of y = f(x) as shown in the above figure Here, we have to search that value of x (x_{min}, say) for which y is minimum at P We start up with a value of x, which is updated iteratively by the Gradient Descent Algorithm According to this algorithm, we update x_{n-1} to x_n as,

$$x_n = x_{n-1} - \eta f'(x_{n-1}) \tag{B 1}$$

Where, η is called the learning rate or the step-size for adjustment of x and $f'(x_{n-1})$ denote the first derivative or gradient of f(x) at $x = x_{n-1}$

Suppose, we are at the point $Q(x = x_{k-1})$ Here, the gradient is negative and thus x is adjusted to x_k through addition of a positive quantity thereby proceeding towards x_{min} Similarly, if we are at $R(x = x_{r-1})$, the gradient becomes positive and the value of x is decreased through subtraction of the gradient term, again proceeding towards x_{min} In effect, we see the solution point (whether at Q or R) to be descenting downhill the function to reach the required solution, thereby justifying the name of the algorithm.

Here, the gradient descent algorithm is explained for one-variable case only. However, this can be extended to higher dimensional spaces also, where we apply the algorithm to reach the lowest point of a multi-dimensional surface. However, the applicability of this algorithm demands the continuity and differentiability of the search surface, which is again preferred to be a quadratic one. In the next section we present the application of Gradient Descent Algorithm to deduce the RBFNN parameter learning rule.

Appendix C

This section focuses on the ***Deductions of the RBFNN Parameter Learning Rules***. The RBFNN parameters include the centres and spreads of the Radial Basis Function nodes and the connection weights between the hidden and the outer layer. We apply the Gradient Descent Algorithm (Described in Appendix B) to deduce the formulae for parameter learning.

Let us consider the generalized RBFNN architecture described in section 2.3.1. For the sake of convenience, we re-describe the same in this section. Figure C.1 shows the generalized RBFNN consisting of N input nodes, P RBF nodes, and M nodes in the output layer.

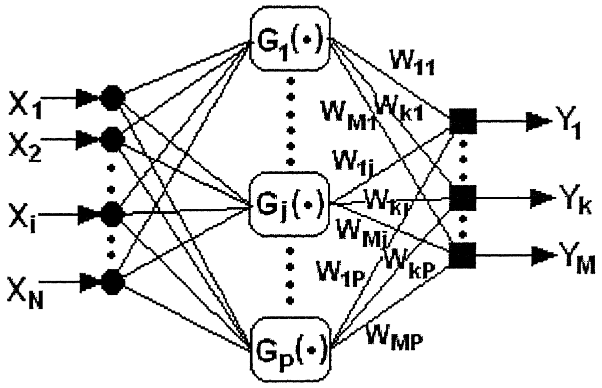


Fig. C.1: Generalized RBFNN Architecture

The N dimensional input vector $X = [\ x_1,..,x_i,..,x_N\ ]^T$ results to the hidden layer activation as $G = [\ g_1,..,g_j,..,g_P\ ]^T$ according to,

$$g_j(X) = \exp\left(-\frac{1}{2}\sum_{i=1}^{N} s_{ij}^{\ 2}(x_i - c_{ij})^2\right) \qquad (C.1)$$

Where, s_{ij} and c_{ij} are the respective i'th components of the spreads and centres of j'th RBF node.

The output vector $Y = [y_1,..,y_k,..,y_M]^T$ is further activated by the hidden layer output as,

$$y_k = \sum_{j=1}^{P} w_{kj} g_j \qquad (C.2)$$

Where, w_{kj} is the weight connecting the k'th Radial Basis Function to the j'th output node.

Let us consider the problem of training this network on a set containing T training patterns. Each of these training patterns consists of the network input vector $X = [x_1,..,x_i,..,x_N]^T$ along with the desired response $D = [d_1,..,d_k,..,d_M]^T$. We initialize the RBFNN parameters by the procedure described in section 2.1.3. However, when the input pattern X is presented to the network, it responds with the output vector $Y = [y_1,..,y_k,..,y_M]^T$. We compute the error in the actual response as,

$$e_k = d_k - y_k \qquad (C.3)$$

Which gives the k'th component of the network response error. Thus we evaluate the net error in network response in the sum-squared manner as,

$$\zeta = \frac{1}{2}\sum_{k=1}^{M} e_k^2 \qquad (C.4)$$

To train the network, we update the parameters with respect to minimization of the error surface ζ. Here, we apply the gradient descent algorithm for the task of minimizing the sum-squared error.

If we fix the learning rate for the weight w_{kj} as η_w, then the weight update rule for the t'th iteration is as,

$$w_{kj}(t) = w_{kj}(t-1) - \eta_w \frac{\partial \zeta}{\partial w_{kj}} \qquad (C.5)$$

From equations C.4 and C.3 and C.2, we can write,

$$\frac{\partial \zeta}{\partial w_{kj}} = e(k)\frac{\partial e_k}{\partial w_{kj}} = -e_k \frac{\partial y_k}{\partial w_{kj}} = -e_k g_j \qquad (C.6)$$

Thus, combining equations C.6 and C.5, we get the final weight update rule as,

$$w_{kj}(t) = w_{kj}(t-1) + \eta_w e_k g_j \qquad (C.7)$$

Similarly, consider the learning rate for the centres to be η_c, then the update rule for c_{ij} in the t'th iteration wll be as,

$$c_{ij}(t) = c_{ij}(t-1) - \eta_c \frac{\partial \zeta}{\partial c_{ij}} \qquad (C.8)$$

Using equations C.4, C.3 and C.2, we proceed to evaluate the gradient term as,

$$\frac{\partial \zeta}{\partial c_{ij}} = \sum_{k=1}^{M} e_k \frac{\partial e_k}{\partial c_{ij}} = -\sum_{k=1}^{M} e_k \frac{\partial y_k}{\partial c_{ij}} = -\sum_{k=1}^{M} e_k w_{kj} \frac{\partial g_j}{\partial c_{ij}} \qquad (C.9)$$

Now, from equation C.1, we can write,

$$\frac{\partial g_j}{\partial c_{ij}} = g_j(-\frac{1}{2}) s_{ij}^2 \{2(x_i - c_{ij})\}(-1) = g_j s_{ij}^2 (x_i - c_{ij}) \qquad (C.10)$$

Thus, combining equations C.8, C.9 and C.10, we get the final expression for centre update as,

$$c_{ij}(t) = c_{ij}(t-1) + \eta_c \left\{ \sum_{k=1}^{M} e_k w_{kj} \right\} \left[g_j s_{ij}^2 \{x_i - c_{ij}\} \right] \qquad (C.11)$$

Proceeding similarly we can also deduce the spread update rule with a learning rate of η_s as,

$$s_{ij}(t) = s_{ij}(t-1) - \eta_s \left\{ \sum_{k=1}^{M} e_k w_{kj} \right\} \left[g_j s_{ij} \{x_i - c_{ij}\}^2 \right] \qquad (C.12)$$

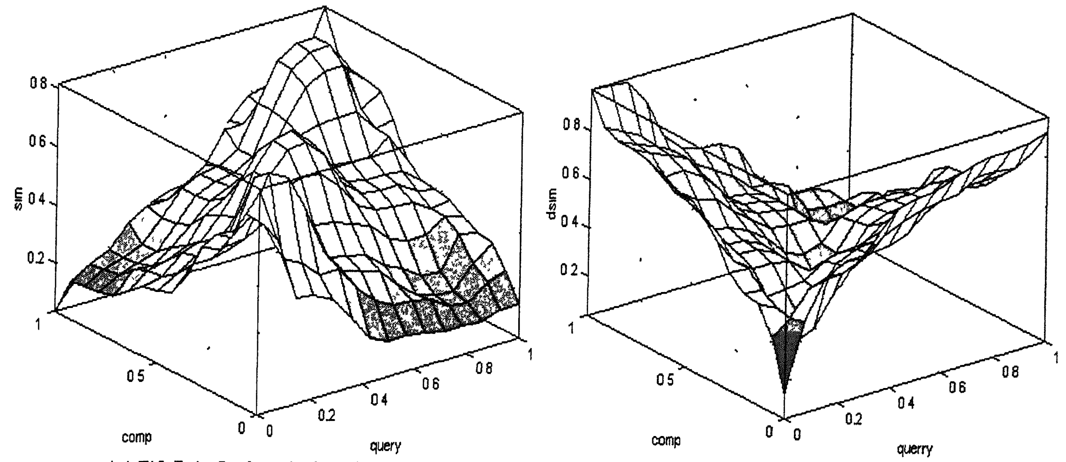
Section 2.3.2 discusses the choices of the learning rates and training procedure in details.

Appendix D

In this section we tabulate the ***FIS rulebases used for the similarity measures***. The antecedent parts Query (Q) and Object of Comparison (C) with AND type of composition are given along the rows and consequents along the columns. The cells of the tables denote the respective weights of the rules. The term set members are abbreviated as VL for "Very Low", M for "Medium" etc. Besides, we also plot the rule surfaces for the respective FIS.

Table D.1 : FIS Rulebase for Line-Line Comparisons

Antecedent		Consequent : Similarity					Consequent : Dissimilarity				
Q	C	VL	L	M	H	VH	VL	L	M	H	VH
VL	VL	0	0	0.1	0.35	0.55	0.7	0.15	0	0	0
	L	0.15	0.2	0.55	0.05	0	0	0.25	0.55	0.1	0.1
	M	0.15	0.5	0.15	0	0	0	0	0.1	0.5	0.4
	H	0.4	0.25	0	0	0	0	0	0	0.3	0.7
	VH	0.5	0	0	0	0	0	0	0	0	1.0
L	VL	0.2	0.35	0.15	0.15	0	0	0.2	0.25	0.5	0.05
	L	0	0	0.05	0.5	0.45	0.65	0.2	0.1	0	0
	M	0.2	0.3	0.35	0.05	0	0	0.2	0.5	0.25	0.05
	H	0.25	0.5	0.05	0	0	0	0	0.05	0.45	0.5
	VH	0.55	0.2	0	0	0	0	0	0	0	1.0
M	VL	0.5	0.2	0	0	0	0	0	0	0.45	0.55
	L	0.25	0.25	0.35	0	0	0	0	0.65	0.1	0.25
	M	0	0	0.1	0.45	0.45	0.55	0.45	0	0	0
	H	0.15	0.2	0.4	0.15	0.05	0	0.25	0.65	0.1	0
	VH	0.2	0.35	0.35	0	0	0	0	0.5	0.35	0.15
H	VL	0.5	0.1	0	0	0	0	0	0	0.1	0.9
	L	0.4	0.4	0.05	0	0	0	0	0.15	0.45	0.35
	M	0.1	0.2	0.55	0.15	0	0	0.25	0.5	0.25	0
	H	0	0	0	0.4	0.6	0.65	0.3	0	0	0
	VH	0	0	0.45	0.35	0.2	0.35	0.45	0.2	0	0
VH	VL	0.45	0.15	0	0	0	0	0	0	0.05	0.95
	L	0.45	0.15	0.2	0	0	0	0	0.1	0.4	0.5
	M	0	0.45	0.35	0.1	0	0	0.15	0.4	0.45	0
	H	0	0.05	0.3	0.5	0.15	0.2	0.6	0.2	0	0
	VH	0	0	0.05	0.3	0.65	0.85	0	0	0	0

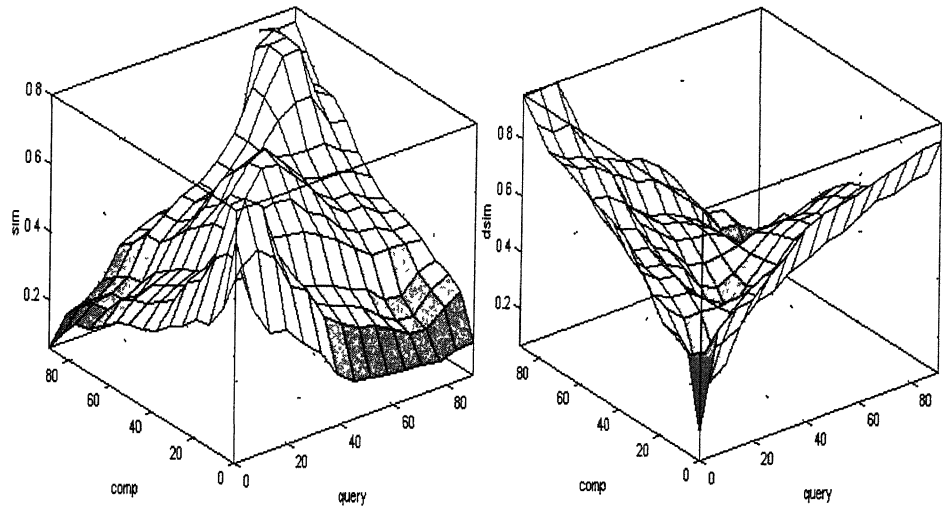


(a) FIS Rule Surface for Line-Line Similarity (b) FIS Rule Surface for Line-Line Dissimilarity

Fig. D.1: FIS Rule Surfaces for Line-Line Comparisons

Table D.2: FIS Rulebase for Angle-Angle Comparisons

Antecedent		Consequent : Similarity					Consequent : Dissimilarity				
Q	C	VL	L	M	H	VH	VL	L	M	H	VH
VL	VL	0	0	0	0.15	0.85	0.75	0.15	0	0	0
	L	0.05	0.15	0.45	0.35	0	0	0.4	0.5	0.1	0
	M	0.3	0.35	0.2	0	0	0	0	0.45	0.45	0.1
	H	0.65	0.15	0	0	0	0	0	0.1	0.3	0.6
	VH	0.75	0.05	0	0	0	0	0	0	0.1	0.9
L	VL	0.3	0.15	0.5	0.05	0	0	0.05	0.8	0.15	0
	L	0	0	0.1	0.3	0.6	0.65	0.25	0.05	0	0
	M	0.1	0.15	0.5	0.15	0	0	0.3	0.55	0.15	0
	H	0.45	0.45	0.1	0	0	0	0	0.05	0.6	0.35
	VH	0.55	0.25	0.05	0	0	0	0	0	0.4	0.6
M	VL	0.5	0.35	0	0	0	0	0	0.05	0.6	0.35
	L	0.15	0.35	0.5	0	0	0	0.25	0.55	0.2	0
	M	0	0	0	0.25	0.75	0.55	0.3	0.1	0	0
	H	0.15	0.1	0.35	0.35	0.05	0	0.25	0.55	0.2	0
	VH	0.3	0.4	0.2	0	0	0	0	0.3	0.45	0.25
H	VL	0.6	0.1	0	0	0	0	0	0	0	1
	L	0.25	0.55	0.05	0	0	0	0	0.05	0.6	0.35
	M	0	0.2	0.5	0.3	0	0	0.3	0.6	0.1	0
	H	0	0	0.05	0.1	0.85	0.75	0.25	0	0	0
	VH	0	0.2	0.3	0.4	0.1	0	0.5	0.35	0.15	0
VH	VL	0.7	0	0	0	0	0	0	0	0	1
	L	0.3	0.5	0	0	0	0	0	0.05	0.45	0.5
	M	0.05	0.45	0.5	0	0	0	0.05	0.7	0.25	0
	H	0	0	0.4	0.35	0.25	0.3	0.5	0.2	0	0
	VH	0	0	0	0.35	0.65	0.5	0	0	0	0



(a) FIS Rule Surfaces for Angle-Angle Similarity (b) FIS Rule Surfaces for Angle-Angle Dissimilarity

Fig. D.2: FIS Rule Surfaces for Angle-Angle Comparisons

Appendix E

This section shows the ***query images and figures of comparison presented to the subjects for obtaining a psycho-visual ranking*** of the same with respect to the queries. The figures consist of rectangles, parallelograms and ellipses as shown below. For the sake of convenience, we also present the ranking results of the psycho-visual tests in table E.1

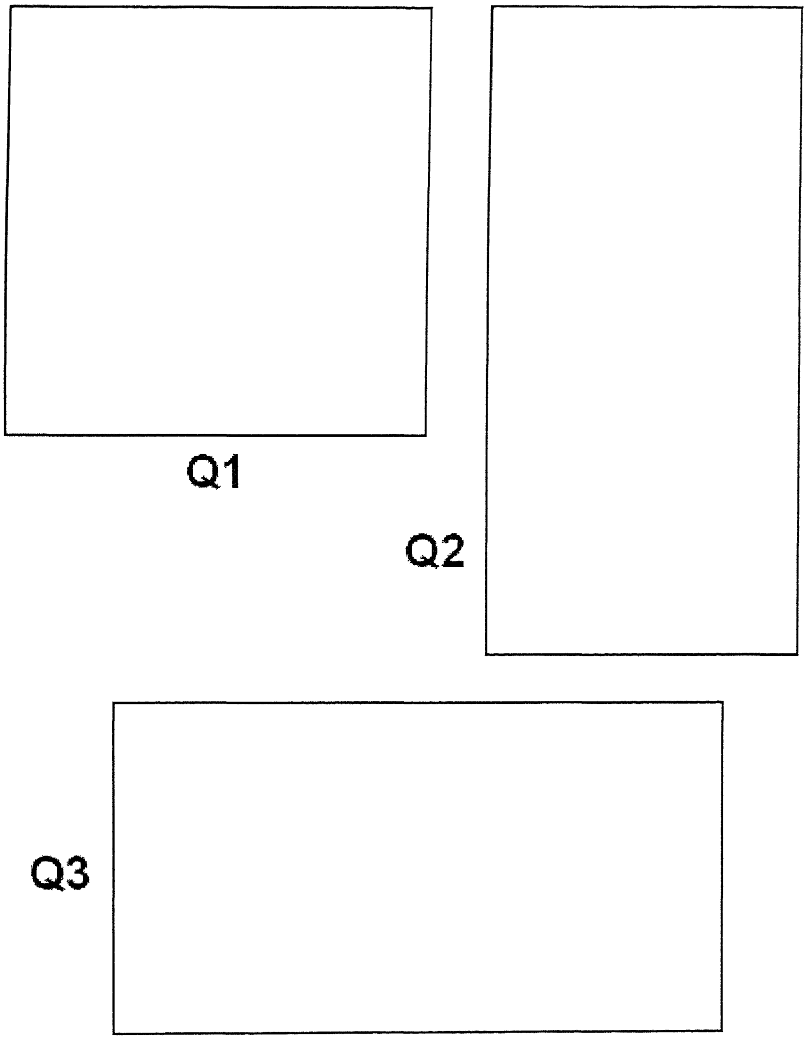


Fig. E.1: Query Images of Rectangles

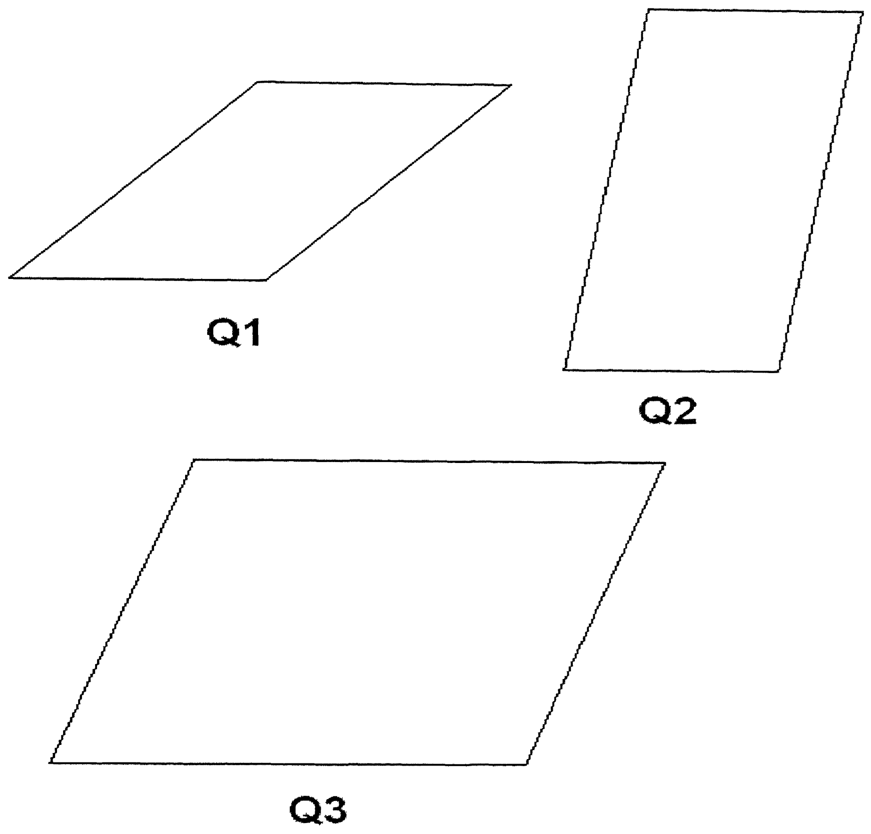


Fig. E.2: Query Images of Parallelograms

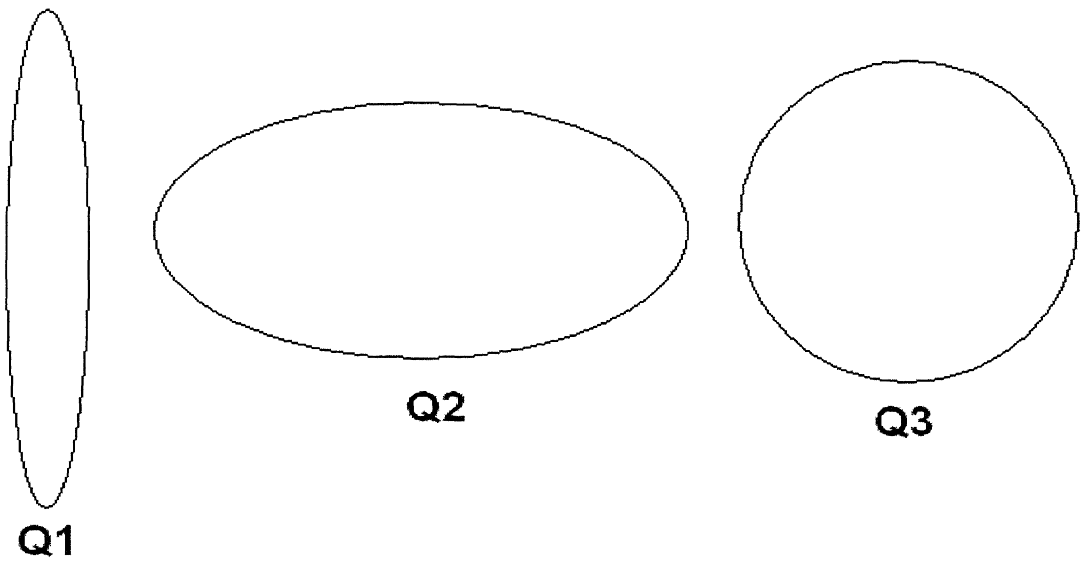


Fig. E.3: Query Images for Ellipses

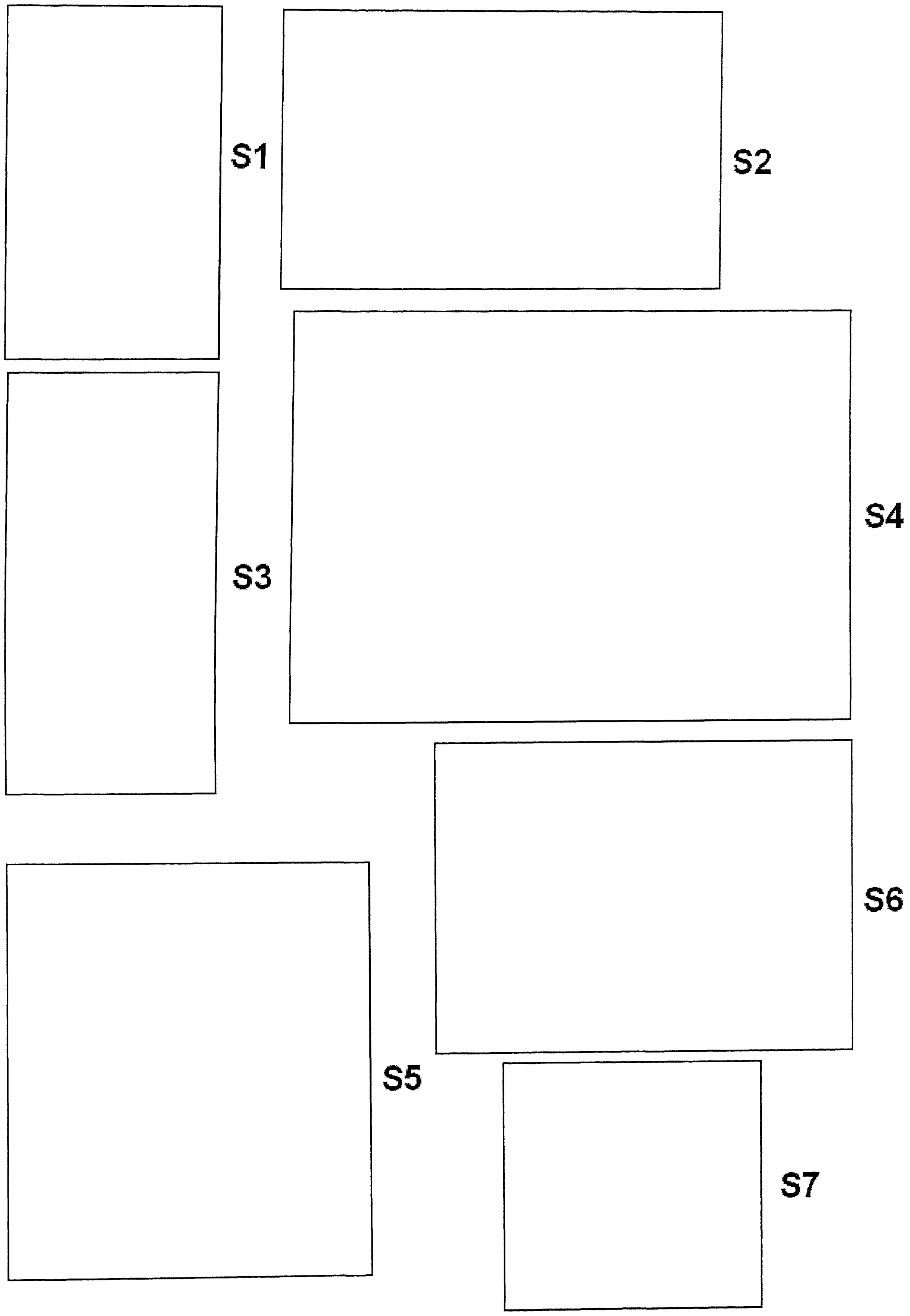


Fig. E.4: Rectangular Figures of Comparison

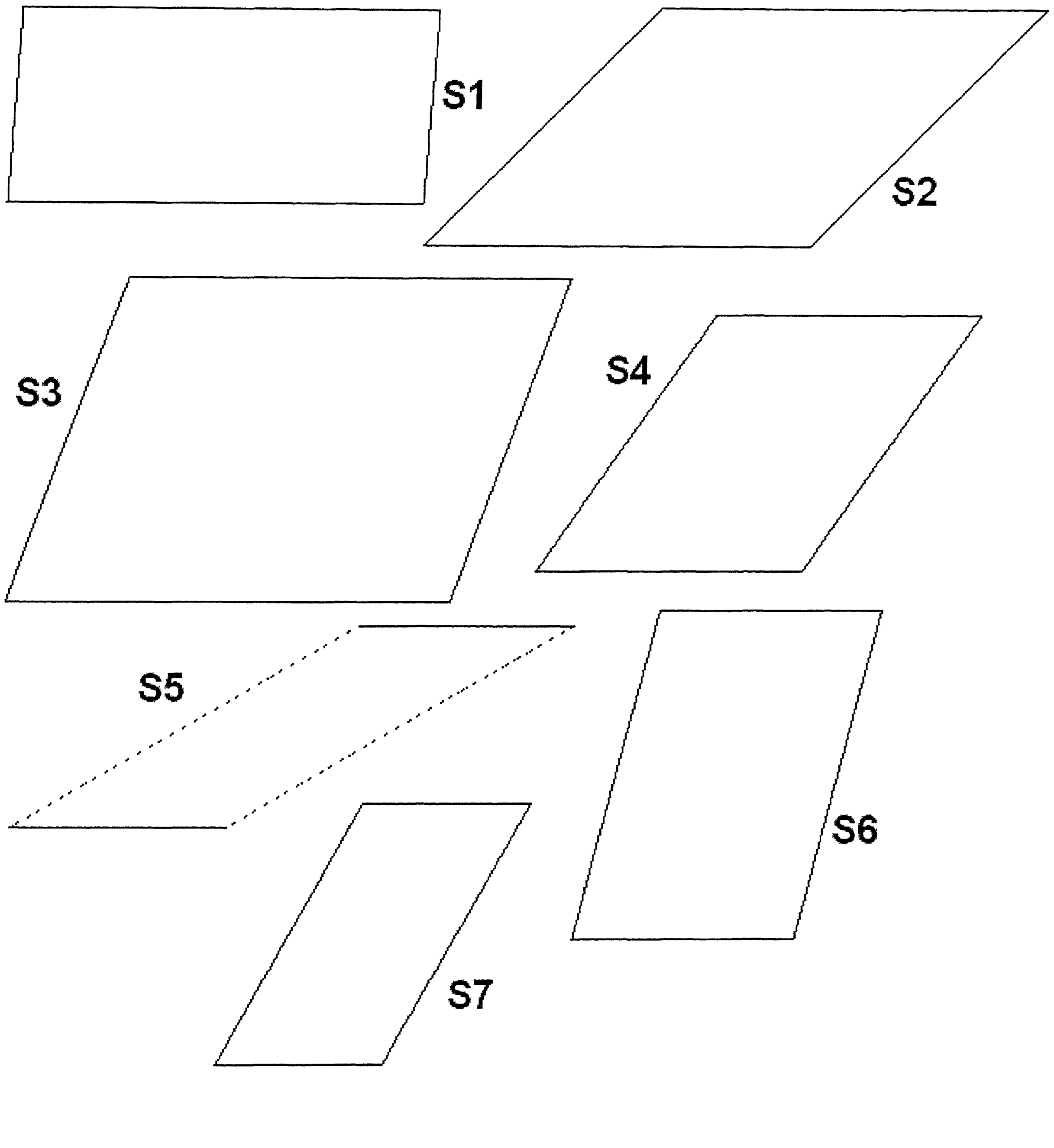


Fig. E.5: Parallelogram Figures Used for Comparison

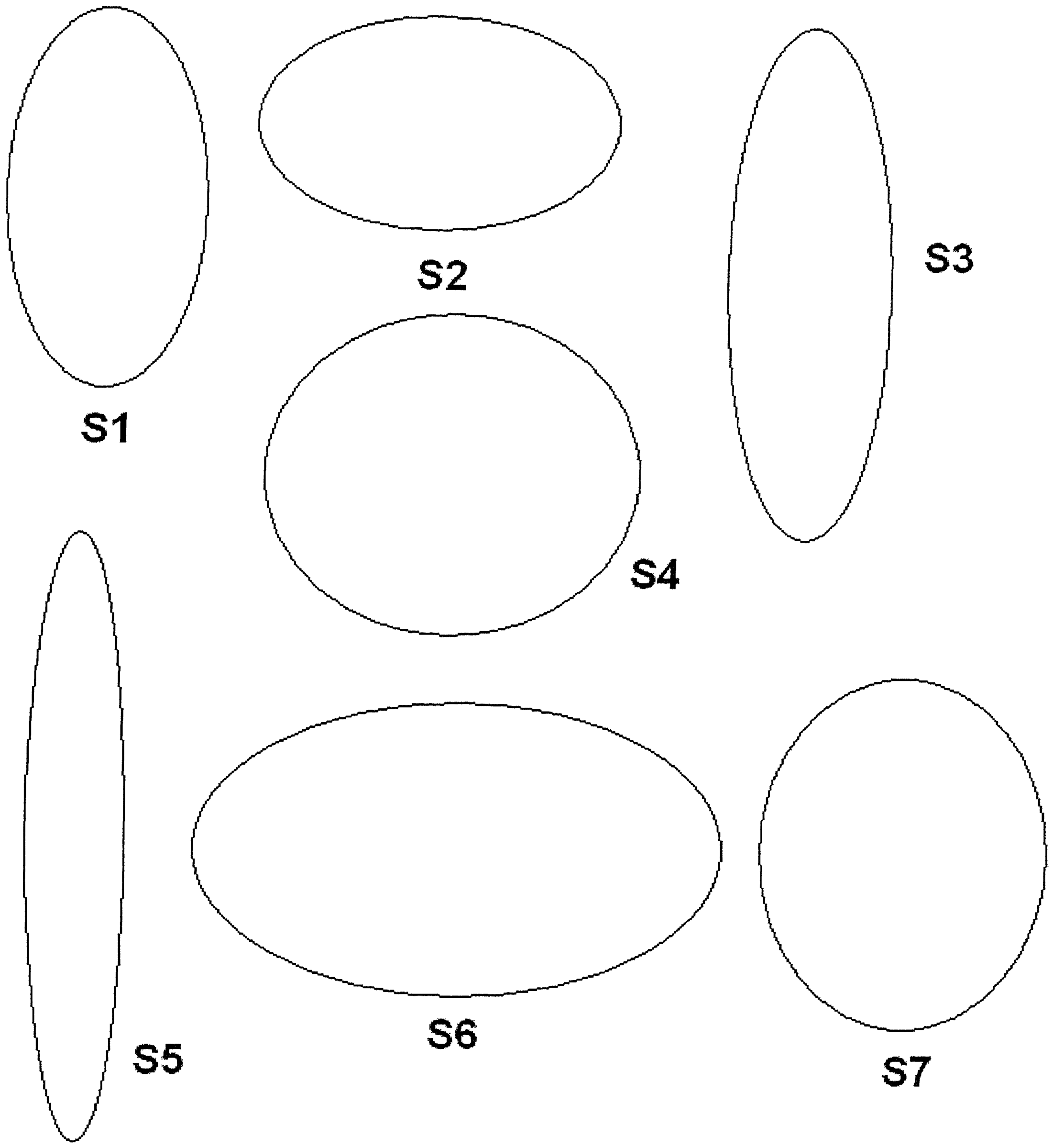


Fig. E.6: Elliptic Figures of Comparison

Table E.1: Results of Psycho-visual Ranking of Geometrical Figures

RANK	1	2	3	4	5	6	7
RECTANGLE							
RQ1	5	6	4	2	7	1	3
RQ2	3	1	5	6	7	4	2
RQ3	2	6	4	5	3	7	1
PARALLELOGRAM							
PQ1	4	7	2	6	5	3	1
PQ2	6	7	4	5	2	3	1
PQ3	3	2	4	6	7	5	1
ELLIPSE							
EQ1	5	3	1	7	4	2	6
EQ2	6	2	4	7	1	3	5
EQ3	4	7	2	6	1	3	5

Bibliography

1. Hon-Son Don, King-Sun Fu, C. R. Liu and Wei-Chung Lin, *"Metal Surface Inspection Using Image Processing Techniques"*, IEEE Transactions of System, Man and Cybernetics, Vol. SMC-14, No.1 January 1984.
2. R.T. Chin and C.A. Harlow, *"Automated Visual Inspection : A Survey"*, IEEE Transactions of Pattern Analysis Machine Intelligence, Vol. 4, pp. 557-573, November 1982.
3. D. Brzakovie and N. Vujoivie, *"Designing a Defect Classification System : A Case Study"*, Pattern Recognition, Vol. 29, No. 8, pp. 1401-141, 1996.
4. Anil K. Jain, *"Fundamentals of Digital Image Processing"*, Prentice Hall, 1989.
5. Jae S. Lim, *"Two Dimensional Signal and Image Processing"*, Prentice Hall, Eanglewood Cliffs, 1989.
6. Ramesh Jain, Rangachur Kasturi, and Brian G. Schunck, *"Machine Vision"*, McGraw-Hill, 1995.
7. Simon Haykin, *"Foundations of Neural Networks"*; Prentice Hall, Upper Saddle River, 1994, New Jersey.
8. Amos Tverskey, *"Features of Similarity"*, Psychological Review, Vol. 84, No. 4, pp. 327-352, July 1977.
9. Karol L. Krumhansl, *"Concerning the Applicability of Geometric Models to Similarity Data : The Interrelationship between Similarity and Spatial Density"*, Psychological Review, Vol. 85, No. 10, pp. 445-463, 1978.
10. Simone Santini and Ramesh Jain, *"Similarity Measures"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 9, pp.871-883, September, 1999.

11. Dennis Dunn and William E. Higgins, *"Optimal Gabor Filters for Texture Segmentation"*, IEEE Transactions on Image Processing, Vol. 4, No. 7, pp. 947-964, July 1995

12. Dennis Dunn William E. Higgins and Joseph Wakely, *"Texture Segmentation using 2-D Gabor Elementary Functions"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 2, pp. 130-149, February 1994

13. Keinosuke Fukunaga, *"Introduction to Statistical Pattern Recognition"*, Academic Press, 1990.

14. Hermann Ney, *"On the Probabilistic Interpretation of Neural Network Classifiers and Discriminative Training Criteria"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 2, pp. 107-119, February 1995.

15. S. Chen, C.F.N. Cowan and P.M Grant, *"Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks"*, IEEE Transactions on Neural Networks, Vol. 2, No. 2, pp. 302-309, March 1991.

16. Clay M. Thompson and Loren Shure, *"Image Processing Toolbox User's Guide"*, The Mathworks Inc., January 1995.

17. F. Kobayashi, M. Tomita, T. Ozeki, *"Calculation of Visual Information of Image"*, Advances in Pattern Recognition and Digital Techniques, Narossa Publishing House, 1999.

18. Barry B. Brey, *"The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, and Pentium Pro Processor - Architecture, Programming, and Interfacing"*, Prentice Hall, 1998.

19. *"MIL-Lite Version 4.0 - User Guide, Command Reference and Board-Specific Notes"*, Manual No. 10514-MU-0205, February 25, 1997

20. David White, Kennard Scribner, and Eugene Olafsen, *"MFC Programming With Visual C++ 6 Unleashed"*, Sams Publishing, 1999.

21. A.C. Bovik, M. Clark, and W.S. Geisler, *"Multichannel Texture Analysis using Localized Spatial Filters"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp. 55-73, January 1990

22. T. Randon and J.H. Husoy, *"Filtering for Texture Classification : A Comparative Study"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 4, pp. 291-310, April 1999

23. David E. Goldberg, *"Genetic Algorithms in Search, Optimization, and Machine Learning"*, Addison-Wesley, 2000

24. K. Deb and R.B. Agarwal, *"Simulated Binary Crossover for Continuous Search Space"*, Journal of Complex Systems, Vol. 9, pp. 115-148, 1995

25. Stephen R. Yhann and Tzay Y. Young, *"Boundary Localization in Texture Segmentation"*, IEEE Transactions on Image Processing, Vol. 4, No. 6, pp. 849-856, June 1995

26. Gregory F. Ashby and Namcy A. Perrin, *"Toward a Unified Theory of Similarity and Recognition"*, Psychological Review, Vol. 95, No. 1, pp. 124-150, 1988.

27. Amos Tversky and Itamar Gati, *"Similarity, Separability, and the Triangle Inequality"*, Psychological Review, Vol. 89, No. 2, pp. 123-154, 1982.

28. Michael Artin, *"Algebra"*, Prentice Hall, 1998

29. Riza C. Berkan and Sheldon L. Trubatch, *"Fuzzy Systems Design Principles – Building IF-THEN Rule Bases"*, IEEE Press, 1997.

30. Kalyanmoy Deb, *"Optimization for Engineering Design : Algorithms and Examples"*, Prentice Hall, 1995.